Chapter 1

The Cloud and Microsoft Azure 101

This chapter focuses on changes that are impacting every organization's thinking regarding infrastructure, datacenters, and ways to offer services. "As-a-Service" offerings—both on-premises and hosted by partners, and accessed over the Internet in the form of the public cloud—present new opportunities for organizations to operate.

Microsoft's solution for many public cloud services is its Azure service, which offers hundreds of capabilities that are constantly being updated. This chapter will provide an overview of the Microsoft Azure solution stack before examining various types of Infrastructure as a Service (IaaS) and how Azure services can be procured.

In this chapter, you will learn to

- Articulate the different types of "as-a-Service"
- Identify key scenarios where the public cloud provides the most optimal service
- Understand how to get started consuming Microsoft Azure services

Understanding the Cloud (or Why Everyone Should Play *Titanfall)*

When I talk to people about Azure or even the public cloud in general, where possible I start the conversation by playing *Titanfall* (`www.titanfall.com`), a game published by Electronic Arts. The game is primarily a first-person shooter, but in addition to running around as a normal person, you get to pilot these massive robots, known as Titans, that are great fun to fight in. Unlike many other games, it is exclusively online and requires a large infrastructure to support the many players. There are many reasons I try to play *Titanfall* when starting my cloud conversations:

- I need the practice, as my teenage son will attest.
- I can write off the console and game because I use it in a business scenario.
- I can present a perfect example of a use case for the public cloud.

Why is *Titanfall* a perfect example of a use case for the public cloud? That is something that will become clear later in this chapter, but in the meantime, I definitely recommend supporting the public cloud and specifically Azure by playing lots of *Titanfall*.

Introducing the Cloud

Every organization has some kind of IT infrastructure. It could be a server sitting under someone's desk, geographically distributed datacenters the size of multiple football fields, or something in between. Within that infrastructure are a number of key fabric (physical infrastructure) elements:

Compute Capacity Compute capacity can be thought of in terms of the various servers in the datacenter, which consist of processors, memory, and other hardware (such as the motherboard, power supply, and so on). I will use the term *compute* throughout this book when referring to server capacity.

Storage A persistent method of storage for data—from the operating system (OS) and applications to pure data such as files and databases—must be provided. Storage can exist within a server or in external devices, such as a storage area network (SAN). SANs provide enterprise-level performance and capabilities, although newer storage architectures that leverage local storage, which in turn replicate data, are becoming more prevalent in datacenters.

Network These components connect the various elements of the datacenter and enable client devices to communicate with hosted services. Connectivity to other datacenters may also be part of the network design. Options such as dedicated fibre connections, Multiprotocol Label Switching (MPLS), and Internet connectivity via a DMZ are typical.

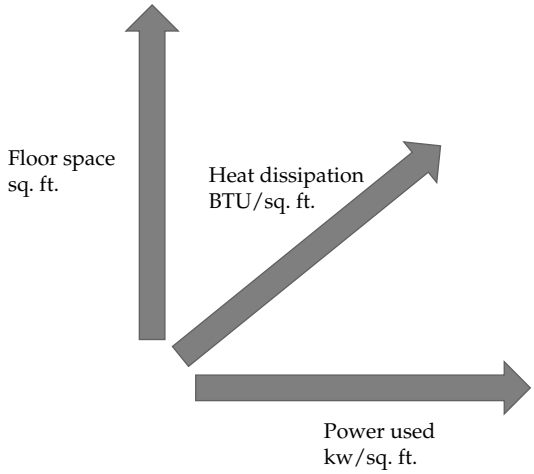
Datacenter Infrastructure An often overlooked but critical component of datacenters is the supporting infrastructure. Items such as uninterruptable power supplies (UPSs), air conditioning, the physical building, and even generators all have to be considered. Each consumes energy and impacts the efficiency of the datacenter as well as its power usage effectiveness (PUE), which provides a measure of how much energy a datacenter uses for computer equipment compared to the other aspects. The lower the PUE, the more efficient the datacenter—or at least the more power going to the actual computing.

Once you have the physical infrastructure in place, you then add the actual software elements (the OS, applications, and services), and finally the management infrastructure, which enables deployment, patching, backup, automation, and monitoring. The IT team for an organization is responsible for all of these datacenter elements. The rise in the size and complexity of IT infrastructure is a huge challenge for nearly every organization. Despite the fact that most IT departments see budget cuts year after year, they are expected to deliver more and more as IT becomes increasingly critical.

Not only is the amount of IT infrastructure increasing, but that infrastructure needs to be resilient. This typically means implementing disaster recovery (DR) solutions to provide protection from a complete site failure, such as one caused by a large-scale natural disaster. If you ignore the public cloud, your organization will need to lease space from a co-location facility or set up a new datacenter. When I talk to CIOs, one of the things at the top of the *don't-want-to-do* list is write out more checks for datacenters—in fact, write out *any* checks for datacenters is on that list.

In the face of increased cost pressure and the desire to be more energy responsible (green), datacenter design becomes ever more complex, especially in a world with virtualization. If the three critical axes of a datacenter (shown in Figure 1.1) are not properly thought out, your organization's datacenters will never be efficient. You must consider the square footage of the actual datacenter, the kilowatts that can be consumed per square foot, and the amount of heat that can be dissipated expressed in BTU per hour.

Figure 1.1
The three axes of datacenter planning



If you get any of these calculations wrong, you end up with a datacenter you cannot fully utilize because you can't get enough power to it, can't keep it cool enough, or simply can't fit enough equipment in it. As the compute resources become denser and consume more power, it's critical that datacenters supply enough power and have enough cooling to keep servers operating within their environmental limits. I know of a number of datacenters that are only 50 percent full because they cannot provide enough power to fully utilize available space.

The Private Cloud and Virtualization

In the early 2000s as organizations looked to better use their available servers and enjoy other benefits, such as faster provisioning, virtualization became a key technology in every datacenter. When I look back to my early days as a consultant, I remember going through sizing exercises for a new Microsoft Exchange server deployment. When sizing the servers required that I consider the busiest possible time and also the expected increase in utilization of the lifetime of the server (for example, five years), the server was heavily over-provisioned, which meant it was also highly underutilized. Underutilization was a common situation for most servers in a datacenter, and it was typical to see servers running at 5 percent. It was also common to see provisioning times of up to six weeks for a new server, which made it hard for IT to react dynamically to changes in business requirements.

Virtualization enables a single physical server to be divided into one or more virtual machines through the use of a *hypervisor.* The virtual machines are completely abstracted from the physical hardware; each virtual machine is allocated resources such as memory and processor in addition to virtualized storage and networking. Each of the virtual machines then can have an operating system installed, which enables multiple operating systems to run on a single piece of hardware. The operating systems may be completely unaware of the virtual nature of the environment they are running on. However, most modern operating systems are enlightened; they are aware of the virtual environment and actually optimize operations based on the presence of a hypervisor. Figure 1.2 shows a Hyper-V example leveraging the VHDX virtual hard disk format.

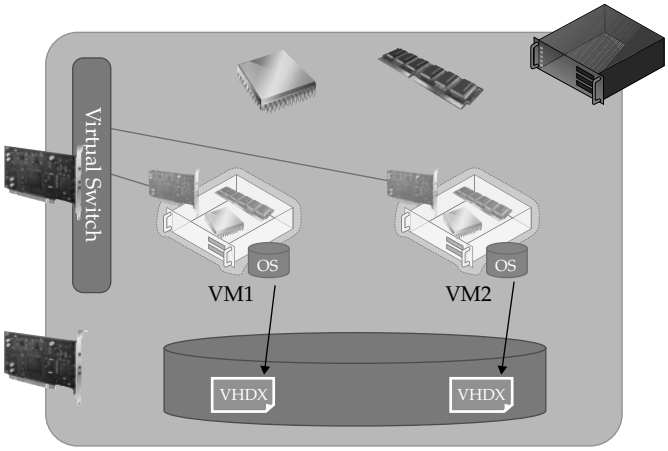


FIGURE 1.2
A high-level view of a virtualization host and resources assigned to virtual machines

Virtualization has revolutionized the way datacenters operate and brought huge benefits, including the following:

High Utilization of Resources Complementary workloads are hosted on a single physical environment.

Mobility of OS Instances between Completely Different Hardware A single hypervisor allows abstraction of the physical hardware from the OS.

Potentially Faster Provisioning Faster provisioning is dependent on processes in place.

High Availability through the Virtualization Solution This ability is most useful when high availability is not natively available to the application.

Simplicity of Licensing for Some Products and OSs For some products and OSs, the physical hardware is allowed to be licensed based on the number of processor sockets, and then an unlimited number of virtual machines on that hardware can use the OS/application. Windows Server Datacenter is an example of this kind of product. There is also an opposite situation for some products that are based on physical core licensing, which do not equate well in most virtualized environments.

There are other benefits. At a high level, if it were to be summed up in five words, I think "more bang for the buck" would work.

The potential of the datacenter capabilities can be better realized. The huge benefits of virtualization on their own do not completely revolutionize the datacenter. Many organizations have adopted virtualization, but have then operated the datacenter as if each OS is still on dedicated hardware. New OS instances are provisioned with dedicated virtualization hosts and even dedicated storage for different projects, which has resulted in isolated islands of resources within the datacenter. Once again, resources were wasted and more complex to manage.

In this book, I'm going to talk a lot about "the cloud." But, for on-premises environments, I would be remiss if I didn't also talk about another big change—the private cloud. Some people will tell you that the private cloud was made up by hypervisor vendors to compete against and stay relevant in the face of the public cloud. Others say it's a revolutionary concept. I think

I fall somewhere in the middle. The important point is that a private cloud solution has key characteristics and, when those are implemented, benefits are gained.

A customer once told me, "Ask five people what the private cloud is, and you will get seven different answers." While I think that is a very true statement, the US National Institute of Standards and Technology (NIST) lists what it considers to be the five critical characteristics that must be present to be a cloud. This applies to both private clouds and public clouds.

On-Demand Self-Service The ability to provision services, such as a virtual machine, as needed without human interaction must be provided. Some organizations may add approval workflow for certain conditions.

Broad Network Access Access to services over many types of networks, mobile phones, desktops, and so on must be provided.

Resource Pooling Resources are organized in a multitenant model with isolation provided via software. This removes the islands of resources that are common when each business group has its own resources. Resource islands lead to inefficiency in utilization.

Rapid Elasticity Rapid elasticity is the ability to scale rapidly outward and inward as demands on services change. The ability to achieve large-scale elasticity is tied to pooling all resources together to achieve a larger potential pool.

Measured Service Clouds provide resources based on defined quotas, but they also enable reporting based on usage and potentially even billing.

The full document can be found here:

`http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`

People often say there is no difference between virtualization and private cloud. That is not true. The difference is the management infrastructure for a private cloud enables the characteristics listed here. To implement a private cloud, you don't need to change your hardware, storage, or networking. The private cloud is enabled through software, which in turn enables processes. You may decide that you don't want to enable all capabilities initially. For example, many organizations are afraid of end-user self-service; they have visions of users running amok and creating thousands of virtual machines. Once they understand quotas and workflows, and approvals, they understand that they have far more control and accountability than manual provisioning provided.

Enter the Public Cloud

The private cloud, through enhanced management processes and virtualization, brings a highly optimized on-premises solution. Ultimately, it still consists of resources that the organization owns and has to house year-round. As I mentioned earlier, CIOs don't like writing checks for datacenters, no matter how optimal. All the optimization in the world cannot counter the fact that there are some scenarios where hosting on-premises is not efficient or even logical.

The public cloud represents services offered by an external party that can be accessed over the Internet. The services are not limited and can be purchased as you consume the service. This is a key difference from an on-premises infrastructure. With the public cloud, you only pay for the amount of service you consume when you use it. For example, I only pay for the amount of storage I am using at any moment in time; the charge does not include the potential amount of storage I may need in a few years' time. I only pay for the virtual machines I need turned on right now; I can increase the number of virtual machines when I need them and only pay for those extra virtual machines while they are running.

Turn It Off!

In Azure, virtual machines are billed on a per-minute basis. If I run an 8-vCPU virtual machine for 12 hours each month, then I only pay the cost for 12 hours of runtime. Note that it does not matter how busy the VM is. You pay the same price whether the vCPUs in the VM are running at 100 percent or 1 percent processor utilization. It's important to shut down and deprovision from the Azure fabric any virtual machines that are not required to avoid paying for resources you don't need. (*Deprovision* just means the virtual machine no longer has resources reserved in the Azure fabric.) The virtual machine can be restarted when you need it again. At that point, resources are allocated in the fabric automatically; the VM will start as expected.

In addition to the essentially limitless capacity, this pay-as-you-go model is what sets the public cloud apart from on-premises solutions. Think back to organizations needing DR services. Using the public cloud ensures there are minimal costs for providing disaster recovery. During normal running, you only pay for the storage used for the replicated data and virtual environments. Only in the case of an actual disaster would you start the virtual machines in the public cloud. You stop paying for them when you can fail back to on-premises.

There are other types of charges associated with the public cloud. For example, Azure does not charge for ingress bandwidth (data sent into Azure—Microsoft is fully invested in letting you get as much data into Azure as possible), but there are charges for egress (outbound) data. There are different tiers of storage, some of which are geo-replicated, so your data in Azure is stored at two datacenters that may be hundreds of miles apart. I will cover the pricing in more detail later in the book, but the common theme is you pay only for what you use.

If most organizations' IT requirements were analyzed, you would find many instances where resource requirements for a particular service are not flat. In fact, they vary greatly at different times of the day, week, month, or year. There are systems that perform end-of-month batch processing. These are idle all month, and then consume huge amounts of resources for one day at the end of the month. There are companies (think tax accountants) that are idle for most of the year but that are very busy for two months. There may be services that need huge amounts of resources for a few weeks every four years, like those that stream the Olympics. The list of possible examples is endless.

Real World Scenario

Super Bowl Sunday and the American Love of Pizza

I'll be up front; I'm English and I don't understand the American football game. I watched the 2006 Super Bowl. After five hours of two minutes of action, a five-minute advertising break, and a different set of players moving a couple of yards, it'll be hard to get me to watch it again. Nonetheless, it's popular in America. As Americans watch the Super Bowl, they like to eat pizza, and what's interesting is the Super Bowl represents a perfect storm for pizza ordering peaks. During the Super Bowl halftime and quarter breaks, across the entire United States, with all four time zones in sync, people order pizza. These three spikes require 50 percent more compute power for ordering and processing than a typical Friday dinnertime, the normal high point for pizza ordering.

Most systems are built to handle the busiest time, so our pizza company would have to provision compute capacity of 50 percent more than would ever normally be needed just for Super Bowl Sunday. Remember that this is 50 percent more than the Friday dinnertime requirement, which itself is much higher than is needed any other time of the week. This would be a hugely expensive and wasteful exercise. Instead Azure is used.

During normal times, there could be 10 web instances and 10 application instances handling the website and processing. On Friday nights between 2 p.m. and midnight, this increases to 20 instances of each role. On Super Bowl Sunday between noon and 5 p.m., this increases to 30 instances of each role. Granted, I'm making up the numbers, but the key here is the additional instances only exist when needed, and therefore the customer is charged extra only when the additional resources are needed. This elasticity is key to public cloud services.

To be clear, I totally understand the eating pizza part!

The pizza scenario is a case of predictable bursting, where there is a known period of increased utilization. It is one of the scenarios that is perfect for cloud computing. Figure 1.3 shows the four main scenarios in which cloud computing is the clear right choice. Many other scenarios work great in the cloud, but these four are uniquely solved in an efficient way through the cloud. I know many companies that have moved or are moving many of their services to the public cloud. It's cheaper than other solutions and offers great resiliency.

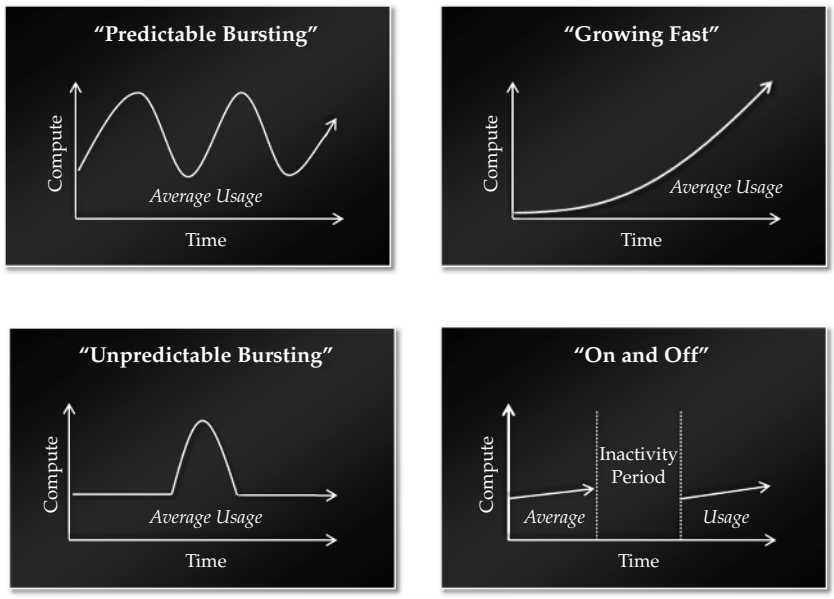


Figure 1.3 The key types of highly variable workloads that are a great fit for consumption-based pricing

In a fast-growing scenario, a particular service's utilization is increasing rapidly. In this scenario, a traditional on-premises infrastructure may not be able to scale fast enough to keep up with demand. Leveraging the "infinite" scale of the public cloud removes the danger of not being able to keep up with demand.

Unpredictable bursting occurs when the exact timing of high usage cannot be planned. "On and Off" scenarios describe services that are needed at certain times but that are completely turned off at other times. This could be in the form of monthly batch processes where the processing runs for only 8 hours a month, or this could be a company such as a tax return accounting service that runs for 3 months out of the year.

Although these four scenarios are great for the public cloud, some are also a good fit for hybrid scenarios where the complete solution has a mix of on-premises and the public cloud. The baseline requirements could be handled on-premises, but the bursts expand out to use the public cloud capacity.

For startup organizations, there is a saying: "fail fast." It's not that the goal of the startup is to fail, but rather, if it is going to fail, then it's better to fail fast. Less money is wasted when compared to a long, drawn-out failure. The public cloud is a great option for startups because it means very little up-front capital spent buying servers and datacenter space. Instead, the startup just has operating expenditures for services it actually uses. This is why startups like services such as Microsoft Office 365 for their messaging and collaboration. Not only do they not need infrastructure, they don't need messaging administrators to maintain it. Public cloud IaaS is a great solution for virtual machines. Once again, no up-front infrastructure is required, and companies pay only for what they use. As the company grows and its utilization goes up, so does its operating expenditure, but the expenditure is proportional to the business. This type of pay-as-you-go solution is also attractive to potential financers, because there is less initial outlay and thus reduced risk.

At the start of this chapter, I said that everyone should play *Titanfall*, and this is where it fits in. *Titanfall* has a large number of artificial intelligence (AI) players, which would be burdensome if their computations had to be performed on a player's console. So, *Titanfall* leverages Azure to provide the services and processing needed. When lots of players are online, an increased number of environments are started in Azure. When fewer players are online, a decreased number of environments are required. Only the environments needed run, thus optimizing costs. The amount of infrastructure that would be required to host something like *Titanfall* would be immense, and leveraging the public cloud presents a perfect-use case, especially when the demand for the service will diminish after a few months as new games are released.

I see the public cloud used in many different ways today, and that adoption will continue to grow as organizations become more comfortable with using the public cloud and, ultimately, trust it. Key use cases today include but are not limited to the following:

Test and Development Test and development is seen by many companies as "low-hanging fruit." It is less risky than production workloads and typically has a high amount of churn, meaning environments are created and deleted frequently. This translates to a lot of work for the IT teams unless the private cloud has been implemented.

Disaster Recovery As discussed, for most companies a DR action should never be required. However, DR capability is required in that extremely rare event when it's needed. By using the public cloud, the cost to implement DR is minimal, especially when compared to costs of a second datacenter.

International DMZ I have a number of companies that would like to offer services globally. This can be challenging—having datacenters in many countries is hugely expensive and can even be politically difficult. By using a public cloud that is geographically distributed, it's easy to offer services around the world with minimal latencies for the end users.

Special Projects Imagine I have a campaign or special analytics project that requires large amounts of infrastructure for a short period of time. The public cloud is perfect for this, especially when certain types of licensing (for example, SQL Server licensing) can be purchased as consumed and other resources are paid for only as required.

A Desire to Get Out of the Datacenter Business I'm seeing more companies that just don't want to maintain datacenters anymore. These organizations will move as much as possible to the public cloud and maintain minimal on-premises infrastructure needed for certain services, such as domain controllers and file and print servers.

Types of Service in the Cloud

Throughout this chapter, I have talked about making services available on-premises with a private cloud and off-premises in the public cloud, but what exactly are these services? There are three primary types of service: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). For each type, the responsibilities of the nine major layers of management vary between the vendor of the service and the client (you). Figure 1.4 shows the three types of service and also a complete on-premises solution. There are many other types of as-a-Service, but most of the other types of services use one of these three primary types. For example, Desktop-as-a-Service really has IaaS as a foundation.

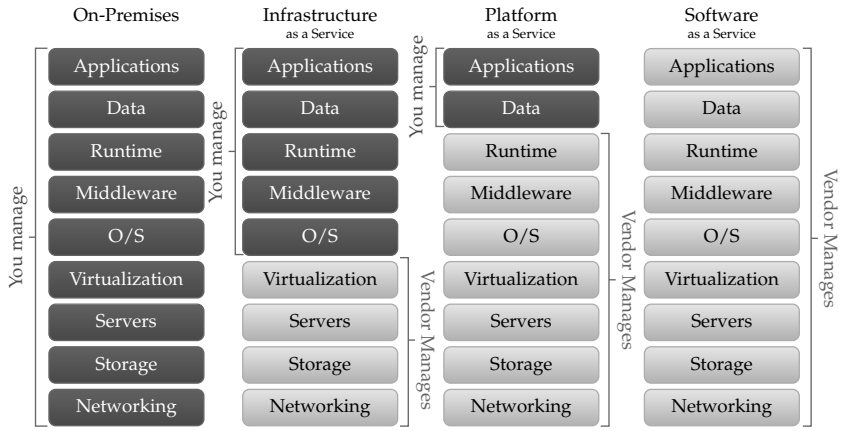


Figure 1.4
The key types of highly variable workloads that are a great fit for consumption-based pricing

IaaS can be thought of as a virtual machine in the cloud. The provider has a virtual environment, and you purchase virtual machine instances. You then manage the operating system, the patching, the data, and the applications within. Examples of IaaS include Amazon's Elastic Computing 2 (EC2) and Azure IaaS, which offer organizations the ability to run operating systems inside cloud-based virtual environments.

PaaS provides a framework where custom applications can be run. Organizations only need to focus on writing the very best application within the guidelines of the platform capabilities, and everything else is taken care of. There are no worries about patching operating systems, updating frameworks, backing up SQL databases, or configuring high availability. The organization just writes the application and pays for the resource used. Azure is the classic example of a PaaS.

SaaS is the ultimate in low maintenance. The complete solution is provided by the vendor. The organization has nothing to write or maintain other than configuring who should be allowed to use the software. Hotmail, a messaging service, is an example of commercial SaaS. Office 365, which provides cloud-hosted Exchange, SharePoint, and Lync services accessed over the Internet with no application or operating system management for the organization, is an enterprise example.

Ideally, for the lowest management overhead, SaaS should be used, and then PaaS where SaaS is not available. IaaS would be used only if PaaS is not an option. SaaS is gaining a great deal of traction with services such as Office 365. PaaS adoption, however, is fairly slow. The primary obstacle for PaaS is that applications have to be written within very specific guidelines in order to operate in PaaS environments. Many organizations have custom applications that cannot be modified. Others don't have the budget to change their applications, which is why IaaS is so popular. With IaaS, an existing virtual machine on-premises can be moved to the IaaS solution fairly painlessly. In the long term, I think PaaS will become the standard for custom applications, but it will take a long time.

IaaS can help serve as the ramp to adopting PaaS. Consider a multitiered service that includes a web tier, an application tier, and a SQL database tier. Initially, all these tiers could run as IaaS virtual machines. The organization may then be able to convert the web tier from Internet Information Services (IIS) running in an IaaS VM and use the Azure web role, which is part of PaaS. Next, the organization may be able to move from SQL running in an IaaS VM to using SQL Azure. Finally, the organization could rewrite the application tier to directly leverage Azure PaaS. It's a gradual process, but the reduced overhead and increased functionality and resiliency at the end state is worth it.

I saw an interesting analogy using the various types of service put in the context of pizza services. (Yes, it's a second pizza example in one chapter; I like pizza.) Take a look at Figure 1.5. No matter where you plan to eat the pizza or how you plan to have it prepared, the actual pizza ingredients are the foundation. Other services and facilities, such as assembling the pizza, having an oven, cooking the pizza, having a table, and serving drinks, are also required. But as we move up the levels of service, we do less and less. At the highest level of service, pizza at a restaurant, we just eat and don't even have to wash up.

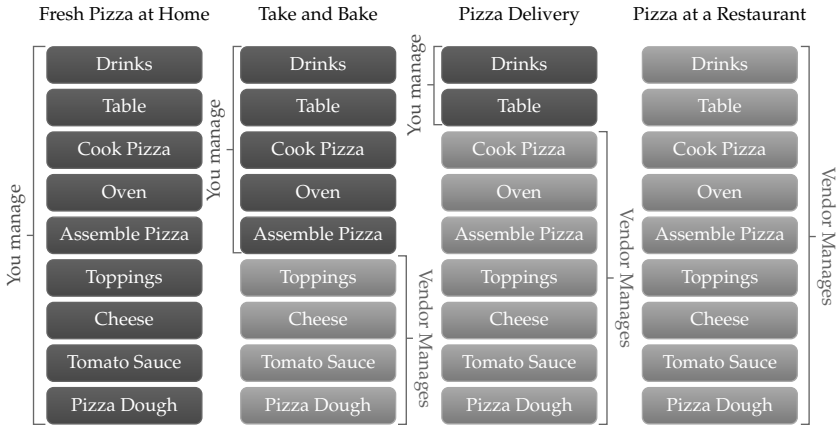


FIGURE 1.5
Various types of Pizza-as-a-Service

The analogy is not perfect. Ideally, I would have had the oven and power as the core fabric. Then, with IaaS, the oven and power would be provided, and I would supply the ingredients, and assemble and cook the pizza (maybe in a pizza cooking class). For PaaS, the dough, sauce, and cheese are provided as a base, and I just add the toppings I want. For SaaS, I eat what I'm given, but only the poshest restaurants can get away with serving whatever they want. I doubt that a pizza restaurant would do well with that model, but you get the idea of the types of service. As you progress through the types of as-a-Service, you are responsible for fewer and fewer elements and can focus on what you care about: the end service/application.

There is another key area in which the pizza analogy is not perfect. In the pizza world, as you progress up the service levels, the service gets better but the total cost increases. When I make a pizza from scratch at home, it's cheaper than eating out at a restaurant. In the IT service space, this is likely not the case. From a total cost of ownership (TCO) for the solution, if I can buy a service like Office 365 as SaaS, that solution is likely cheaper than operating my own Exchange, SharePoint, and Lync solution on-premises when you consider the server infrastructure, licenses, IT admin, and so on.

Microsoft Azure 101

Microsoft has many solutions in the public cloud. The IaaS and PaaS services focus on Azure services, formerly known as Windows Azure, but rightly renamed, as many services are not Windows-centric. Although the focus of this book is on services related to infrastructure, primarily IaaS, this section will provide a high-level overview of all the key Azure services. It's important to fully understand the capabilities that are available.

Figure 1.6 shows the three main building blocks of the Azure platform along with the networking and traffic management services that enable various types of connectivity. Azure Compute provides the primary compute capabilities, including virtual machines, cloud services, websites, plus the fabric controller, which manages all the virtual machines and hosts that provide the Azure platform. Azure Data Services, as the name suggests, provides storage, backup, and the SQL Server capabilities in the cloud, including relational databases (which are not available in the core Microsoft Azure component). Finally, the Azure App Services provide services such as access control, caching, directory services, and a service bus for communication between components within and outside of Azure. An Azure Marketplace facilitates the buying and selling of Azure applications, components, and datasets.

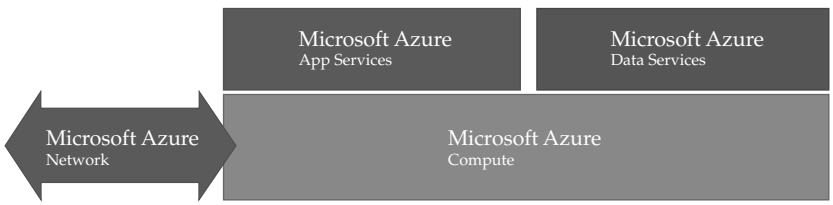


Figure 1.6
The three main building blocks of the Azure Platform: Azure, App Services, and Data Services

Once you decide what servers, storage, load balancing, and other goodness is needed to run your applications, you need to decide where you want those capabilities hosted. Microsoft has many datacenters distributed throughout the world; Azure applications can run from any of them. There are currently four datacenters in the United States, and two each in Europe and

Asia regions. When an application is deployed to Azure, the customer can select where the application should be deployed. All Azure datacenters are paired to provide protection from site failure and data is replicated.

In the next sections, I will introduce you to many of the services available in Azure. Understand that it is a constantly changing service, and my focus is to simply provide you with an idea of the scope of services. For the most up-to-date list of available services and to get more details, visit `http://azure.microsoft.com/en-us/services/`. Those services I will be covering in detail throughout the book, such as Azure Active Directory, Azure Automation, Azure Traffic Manager, and of course the networking, won't be covered in this chapter. (I have another 400 or so pages to tell you about those.) Microsoft often releases new services in a preview mode. You need to apply for preview services before you can use them. To check to see the previews that are available and to sign up, use the Preview website here:

```
http://azure.microsoft.com/en-us/services/preview/
```

Microsoft Azure Compute

The main building block of the Azure platform is Microsoft Azure itself. Here you'll find the key capabilities that enable the cloud-based hosting of applications and data. As Azure has evolved, so too has the naming and hierarchy of services. If you were able to look at what Azure was a year ago, the components would seem different from those I'm describing today.

The virtual machine, as shown in Figure 1.7, is the part that actually runs the applications, which could be a website, some custom middleware code, or some legacy application. All of the compute capabilities are enabled through virtual machines, which vary in size. Although virtual machines are directly accessible and used with Azure IaaS, other services, such as PaaS, websites, networking, and so on, are built using virtual machines; they simply may not be visible to you. I will focus on the IaaS virtual machines in this chapter.

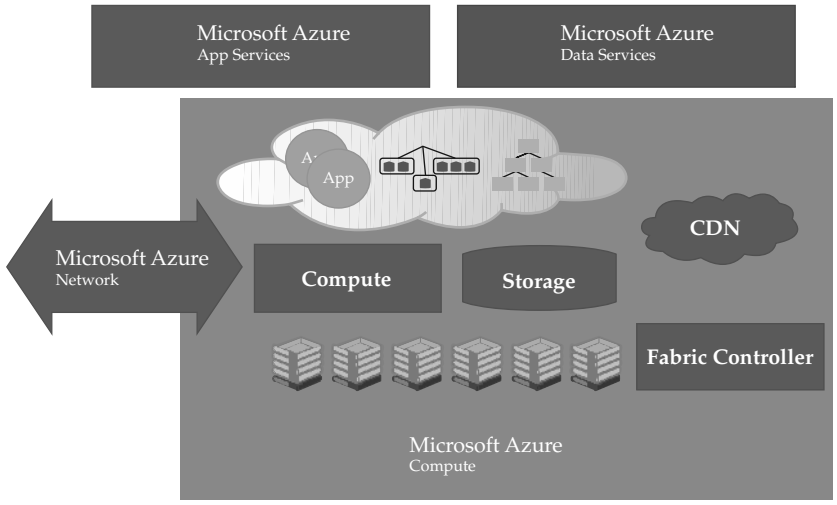


Figure 1.7
The main components that make up Microsoft Azure Compute

A web role acts as the web server for your web applications, such as ASP.NET, Classic ASP, PHP, Python, and Node.js. The web role leverages IIS to run the web applications. If you request five instances of a website role for your web application, then behind the scenes, five virtual machines running IIS are created and load balanced, and all run your web code. If in the future you want additional instances, you just request additional instances, and Azure automatically creates new instances of the web role, deploys your custom code, and adds those new instances to the load balancing. Azure websites, which are separate from the web roles, provide a very fast way to deploy a web application.

A worker role is used to run tasks for backend applications that are not IIS web applications but leverage PaaS. As with the web role, when you deploy your application, you tell Azure how many instances of the role you want, and Azure distributes your application to all instances and balances load. Using worker roles, you can run applications such as Java Virtual Machines or Apache Tomcat. This is where the Azure flexibility shines.

You can use any combination of web roles, worker roles, and VMs to support your application. Some applications may only require web roles, some may need web and worker roles, and some can be deployed using just VMs. The point is the flexibility is there to create roles that meet the needs of the application you are deploying.

Azure Compute also features a Mobile Services set. These services are designed to provide the backend for mobile applications running on Windows, iOS, and Android platforms. Numerous services are available; some of the most useful allow integration into authentication services, such as Microsoft and Facebook accounts, and provide the ability to push notifications to devices.

An auto-scale capability allows multiple Azure virtual machines to be grouped together using a construct known as an *availability set*. Through configuration, you can specify the minimum and maximum number of virtual machines that should run and how scaling should be performed. For example, if CPU requirements in each VM exceed a certain amount, scaling should occur. To accomplish this, you must precreate all the possible virtual machines in the set. The Azure auto-scale then simply stops (deprovisions) and starts them as needed and optimizes your charges so that only the VMs needed are running. As of this writing, auto-scale will not automatically create new instances of virtual machines from a template. Azure does make it easy to request and instantly deploy additional instances of a role. Auto-scale also allows you to leverage the System Center App Controller, and you can programmatically request new instances by writing your own auto-scaling functionality.

The Azure fabric controller itself seems to work magic. As a customer, you can deploy your application, and Azure just spins up as many instances as you say. You can scale up or down at any time. Your service is always available per the Azure "99.95%" monthly service level agreement (SLA). The operating systems and dependent services are constantly patched and tuned. The magic is enabled by the Azure fabric controller, which itself is a distributed application running on Azure that has a fabric agent that runs on all the virtual machines (except IaaS VMs) and hosts that make up the Azure compute fabric. The fabric controller constantly monitors, and if it sees a problem, it can spin up new instances of a role. If a request is made for more instances of a role, the fabric controller creates the new instances and adds them to the load balancer configuration. The fabric controller handles all patching and updates (again, except those VMs that are IaaS). Patching and updating is a key reason that Azure applications can be covered by the "99.95%" SLA. You must deploy at least two instances of any role to take advantage of this capability, as the fabric controller will take down one instance, patch it, and then once that instance is running again, take down the other. As you have more and more instances, more instances can be patched simultaneously by grouping role instances

in upgrade domains. When patching occurs, all instances within an upgrade domain are brought down and updated at the same time. Then, once complete, the next upgrade domain is updated, and so on.

Microsoft Azure Data Services

The ability to store data is critical to any service. Azure data services provide several types of storage and make them available to both Azure-based services and on-premises solutions.

Four primary types of storage are available:

Binary Large Object (BLOB) Azure offers an unstructured collection of bytes that can be used to store basically anything, including large media files. Currently BLOBs can scale up to 200 TB.

Tables Table storage can be confusing because these are not tables in the relational table sense. For relational database needs, Azure uses a SQL database. Tables are a structured store based on key-values. They are designed to store large amounts of data for massive scale where some basic structure is required, but relationships between data don't need to be maintained. Azure tables can be thought of as a NoSQL implementation. These constitute a growing class of database management systems that don't use SQL as their language or implement relational table capabilities.

Queues Queues primarily provide reliable and persistent messaging between applications within Azure. A particularly common use for queues is for the communication between web roles and worker roles. Queues have a basic functionality, which makes them fast. They don't have familiar characteristics, such as first in, first out (FIFO). Instead, developers implement their own features on top of the Azure queue feature.

Files This is an SMB 2.1 implementation that provides an easy method to share storage within an Azure region. Using the standard SMB protocol, files can be created and read. The reads and writes are being implemented by Azure Storage.

Azure Drive is a feature that allows a BLOB to be used as a virtual hard disk (VHD) and formatted as a NTFS volume. Although it allows applications to interact with the BLOB as a volume, it is not actually a different type of storage.

Any data stored in Azure is replicated three times within the same datacenter, and any Azure BLOB, table, and file content are also geo-replicated to another datacenter hundreds of miles away to provide resiliency against site-level disasters. The geo-replication is not synchronous, but it is performed very quickly. There should not be much lag between the data content at the primary location and the geo-replicated location. Read-access is available to the geographically replicated copy of the storage if required. Applications interact with storage using HTTP or HTTPS. For the tables, the OData (Open Data Protocol) is used. OData builds on web technologies to provide flexible ways to interact with data.

Microsoft also features an import/export capability that provides a clean way to transport large amounts of data where transportation over the network is not practical. The import/export service copies data to a 3.5-inch SATA HDD that is encrypted with Bitlocker. The drive is then shipped to the Microsoft datacenter, where the data is imported and made available in your Azure account.

Where a relational database capability is required, Azure SQL databases, which provide relational data through a subset of SQL Server capability in the cloud, should be used. This gives the Azure application full access to a relational database where needed. SQL Azure is a

pricing model separate from the Computer and Storage components of Azure, because you are paying for the SQL service rather than raw storage. Two types of database are available: Web Edition (10 GB maximum database size) and Business Edition (150 GB maximum database size). Billing is based on database size in gigabyte increments. SQL Reporting is also available.

Other types of service are available as well. For your Big Data Azure features, HDInsight is a Hadoop-based service that brings great insight into structured and unstructured data. A shared cache service is available to provide improved storage performance. Another service that is gaining traction is Azure Backup, which provides a vault hosted in Azure to act as the target for backup data. Data is encrypted during transmission and encrypted again when stored in Azure. This provides an easy-to-implement, cloud-based backup solution. Currently, Windows Server Backup and System Center Data Protection Manager can use Azure Backup as a target. Azure Site Recovery, which allows various types of replication to Azure, also falls within the Data Services family of services.

Microsoft Azure App Services

Azure App Services encompass various technologies that can be used to augment Azure applications. As of this writing, a number of technologies make up the Azure App Services, including the following:

Content Delivery Network (CDN) Azure datacenters are located all around the globe, as we've already discussed, but there are certain types of data you may want to make available even closer to the consumer for the very best performance of high-bandwidth content. The CDN allows BLOB data within Azure Storage to be cached at points of presence (PoPs). PoPs are managed by Microsoft and are far greater in number than the Azure datacenters themselves. Here's how the CDN works. The first person in a region to download content would pull the data from the CDN, which originates from the Azure Storage BLOB at one of the major datacenters. The content is then stored at that CDN PoP, and from there, the data is sent to the first user. The second person to view the data in that location would then pull the data directly from the PoP cache, thus getting a fast response. Use of the CDN is optional and has its own SLA with a pay-as-you-go pricing structure based on transactions and the amount of data. Many organizations leverage the CDN for delivering their high-bandwidth data, even if it's separate from an actual Azure application. CDN is easy to enable.

Microsoft Azure Active Directory Active Directory (AD) provides an identity and access management solution that integrates with on-premises (where required) and is leveraged by many Microsoft solutions, such as Office 365, in addition to your own custom solutions. Multifactor authentication is available and can enable your mobile phone to act as part of the authentication process; a code required to complete the logon can be sent in a text.

Service Bus Service Bus supports multiple messaging protocols and provides reliable message delivery between on-premises and cloud systems. Typically, problems occur when on-premises, mobile, and other solutions attempt to communicate with services on the Internet because of firewall and IP address translation. With Azure Service Bus, the communication is enabled through the Service Bus component.

Media Services Providing high-quality media experiences, such as streaming of HD live video, is the focus of media services. Various types of encoding and content protection are supported.

Scheduler As the name suggests, the scheduler queues jobs to run on a defined schedule.

Reliable vs. Best-Effort IaaS

I want to cover how a service is made highly available early on in this book; it's a shift in thinking for many people, but it's a necessary shift when adopting public cloud services.

For most datacenter virtual environments, on-premises means that the infrastructure is implemented as a reliable infrastructure, as shown in Figure 1.8. Virtualization hosts are grouped into clusters, and storage is provided by enterprise-level SANs. A stand-alone virtual machine is made reliable through the use of clustering. For planned maintenance operations such as patching, virtual machines move between nodes with no downtime through the use of technologies like Live Migration. The result is minimal downtime to a virtual machine.

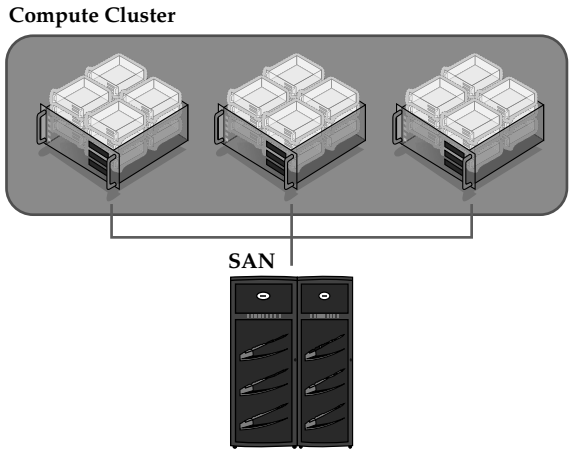


FIGURE 1.8
A reliable on-premises virtual environment

This type of reliable infrastructure makes sense for on-premises, but it does not work for public cloud solutions that have to operate at mega-scale. Instead, the public cloud operates in a best-effort model. Despite the name, best effort does not mean it's worse—in reality, it often is better. Instead of relying on the infrastructure to provide reliability, the emphasis is on the application. This means always having at least two instances of a service and organizing those services in such a way as to assure that those two instances do not run on the same rack of servers (fault domain). Further, the two instances can never be taken down at the same time during maintenance operations. The application needs to be written in such a way that multiple instances are supported. Ideally, the application is aware there are multiple instances and can perform certain types of recovery action where required. (See Figure 1.9.) Some kind of load-balancing technology is also needed so that incoming connections are sent to an active instance.

The reality is with a multiple-application instances model, the level of application availability is higher than when using a reliable infrastructure model. Although the reliable infrastructure provides zero downtime from planned maintenance operations (as does the best-effort infrastructure with multiple instances of the application on different fault domains), a reliable infrastructure cannot protect from downtime in an unplanned failure. If a host fails, the cluster

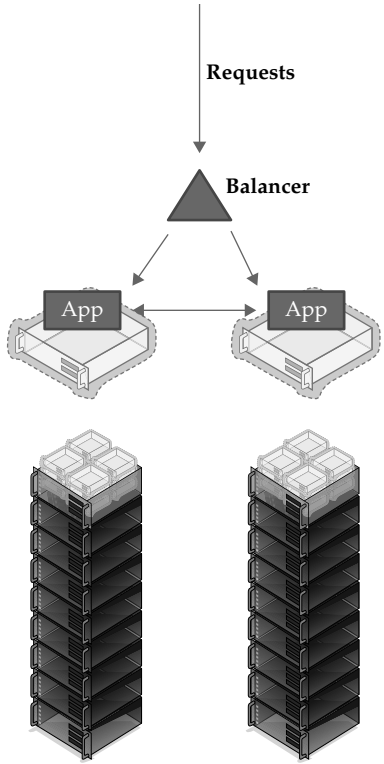


Figure 1.9
An example of design in a best-effort infrastructure

infrastructure in the reliable model will restart the VM on another node. That restart takes time, and the OS and/or application may need integrity checks and possibly recovery actions because essentially the OS inside the VM was just powered off. In the best-effort infrastructure using multiple instances of the application, unplanned failure is covered without risk of corruption, as the other instance would still be running. Additionally, though the reliable infrastructure protects against host failure, it cannot provide protection if the OS inside a VM hangs or errors in some way that is not seen as a problem for the host environment. In such a case, no corrective action is taken, and the service is left unavailable. Using an application-aware approach, any issues in the application could be detected, and clients would be sent to the other instance until any problems were resolved.

Getting Access to Microsoft Azure

For a typical on-premises solution, there are many hurdles to the access and adoption of a new technology. You need hardware to run the solution and somewhere to store that hardware, and you have to obtain and deploy the various operating system requirements and applications. With public cloud solutions, the services are sitting out there, just waiting for you to start using them. Primarily, that means a way to pay for the services. If, however, you want to just try Azure, you don't even need that.

Importance of Planning for Services

Azure services (like most public cloud services) are easy to access and can be used with almost no barriers. This does not mean an organization should instigate the use of public cloud services with any less consideration and planning than would be given to solutions implemented on-premises. In fact, because the services are hosted externally, additional planning is likely required to ensure integration with on-premises solutions and adherence with security and compliance requirements.

Many organizations (or more specifically parts of organizations) have not done this planning and adopted public cloud services without central governance or planning, which causes problems in the long run. It is common to hear about a particular business unit in a company using the public cloud because their own internal IT department takes too long to deliver a service. Essentially, that business unit makes a decision to host the solution themselves without the required skill sets to ensure the solution is secure and adheres to requirements. The public cloud offers huge benefits, but ensure that its adoption is well thought out.

Free Azure Trials and Pay-as-You-Go

The first way to gain access to Azure (and the best way for most people who want to get an idea of what it's like to use Azure) is to sign up for a one-month free trial. The trial includes a $200 Azure credit that can be used for any Azure services. The free trial offer is available here:

```
http://azure.microsoft.com/en-us/pricing/free-trial/
```

To sign up, you will need a Microsoft Account (formerly known as a Windows Live ID), a phone number, and a credit card. Nothing is charged to the credit card, nor will anything be charged to the credit card—by default, once you hit the $200 Azure spend, the account is suspended. You have to agree to pay for services beyond the included $200 and change the default configuration to a Pay-as-You-Go subscription before any expense is incurred. A credit card is required for identity verification only. Once the 30 days has expired, again by default, the account is deactivated and all services removed unless you have converted the account to Pay-as-You-Go. Of course, you can always simply buy Azure services on a Pay-as-You-Go basis and be billed for the service used at the end of each billing cycle.

As you can imagine, any kind of public cloud service where VMs can be run for free is attractive to people with dubious intentions, such as running botnet services and even mining crypto currencies. Microsoft, like all public cloud services, tries to ensure trial services are used in the spirit they are intended: to try out Azure.

A Bucket of Azure Money

Unlike many other types of purchasing in the IT world, with Azure you essentially have a bucket of money to use for Azure services. You can use that bucket for any of the types of service, virtual machines, storage, media services, backup, databases—it doesn't matter. You don't purchase $10,000 of VM quota and $5,000 of storage. You purchase $15,000 of Azure service and then spend it however you want. This is a much better option for organizations that, over time, may change the type of Azure service they want. You may begin by running SQL Server in Azure IaaS VMs with lots of storage but eventually move to using Azure SQL Database. You can make that change easily, since Azure money is not service specific.

Azure Benefits from MSDN Subscriptions

Another great way to experiment with Azure (and even use it on an ongoing basis as part of development and testing) is to leverage the Azure benefits that are part of Microsoft Developer Network (MSDN) subscriptions. MSDN subscriptions are a paid service that enables access to pretty much all Microsoft software. It is intended to be used as part of development and testing efforts. Also included with MSDN subscriptions is a monthly Azure credit, which varies depending on the level of the MSDN subscription, as shown in Table 1.1. (The credits are accurate as of this writing, but they could change over time.) Additionally, the MSDN Azure credits go further than regular Azure spending since the OS licenses are part of the MSDN subscription. Windows virtual machines are 33 percent discounted, as are some other services. Benefits for MSDN Ultimate subscriptions are documented on the MSDN Azure benefits details page:

```
http://azure.microsoft.com/en-us/offers/ms-azr-0063p/
```

Table 1.1: MSDN Azure benefits

Subscription Level	Visual Studio Professional with MSDN	Virtual Studio Test Professional with MSDN	MSDN Platforms	Visual Studio Premium with MSDN	Visual Studio Ultimate with MSDN
Azure Credits per month	$50	$50	$100	$100	$150

http://azure.microsoft.com/en-us/pricing/member-offers/msdn-benefits-details/

Like the Azure trial accounts, by default the MSDN Azure benefit accounts will not allow you to exceed the Azure credits associated with your MSDN subscription. When your monthly limit is reached, your services will be stopped until the start of the next billing month. You need to manually disable the spending limit and specify a credit card if you wish to use more than the Azure benefit credits each month. With the highest-level MSDN subscription (Ultimate), it's possible to run three standard-service single-core virtual machines with just under 2 GB of memory all month, which is a pretty nice testing environment. It is important to note that the MSDN Azure benefit is for development and testing only and cannot be used to run production services. Microsoft can shut down your services if it finds they are being used for production purposes.

Most Microsoft-focused developers already have MSDN subscriptions, so this is a great way to get Azure services to continue that development into the cloud. When I talk to customers about the MSDN benefit, a common question I hear is: "We have lots of MSDN subscriptions—can we pool all the Azure credits together to use as an organizational credit pool?" The answer is no, there is no way to pool MSDN Azure credits together—they are designed to be used by the individual MSDN subscriber for their test and development efforts.

As Figure 1.10 shows, it's easy to check on the current credit status of your subscription; by default, it is pinned to the Azure Startboard. If you select the pinned tile, more details related to the billing (for example, the number of days left) appear. Detailed information is also available on the older portal available at `https://manage.windowsazure.com`.

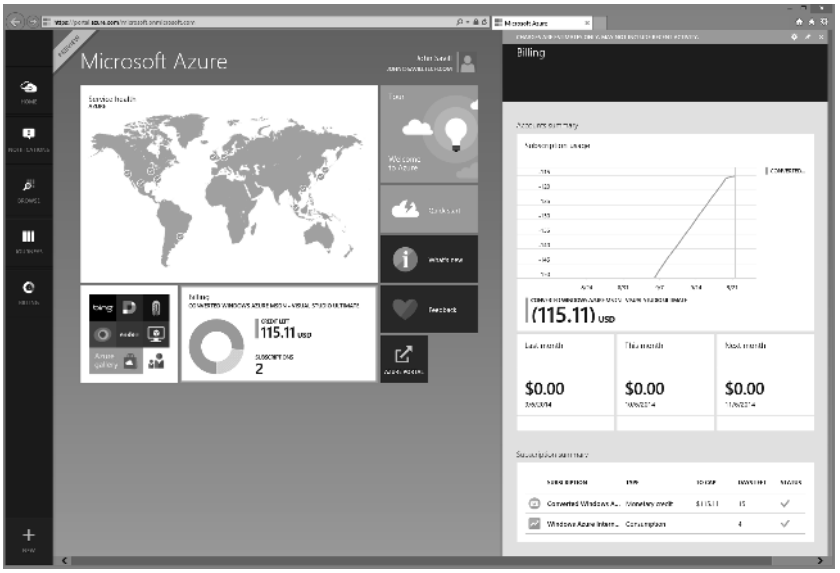


FIGURE 1.10
Viewing billing information for Azure subscription

If you wish to remove the spending limit and pay for Azure services beyond the MSDN subscription, perform the following steps:

1. Navigate in a browser to `https://account.windowsazure.com/Subscriptions/`.
2. Select the MSDN Azure subscription that has a spending limit enabled. Click the subscription name.
3. In the Subscription Status section, click the Remote Spending Limit link, shown in Figure 1.11.
4. When the Remote Spending Limit dialog box opens, change the selection to Yes, remote the spending limit, and then click the check mark.

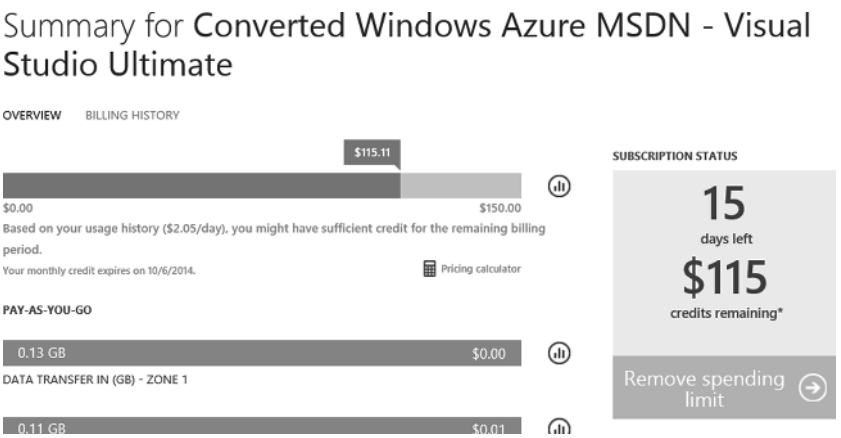


FIGURE 1.11
Removing the spending limit for Azure subscription

It's important you understand the ramifications of removing the spending limit. If you accidentally start a lot of virtual machines and leave them running, you may end up with a large bill at the end of the month. So, if you do remove the spending limit, ensure that you keep a close eye on your Azure spend.

Azure Open Licensing

Another option for organizations that don't want to use the Pay-as-You-Go via a credit card but are not ready for an Enterprise Agreement Azure commitment is to purchase Azure Open Licensing credits from a distributor or reseller. The Azure Open Licensing credits model is similar to that of a prepaid phone. You buy credits and add minutes or services to your phone. Then, you add more credits when you get low. Azure Open Licensing has the same credit model. Units of Azure credits are purchased in blocks equivalent to $100 USD. At purchase, an Online Service Activation (OSA) key is provided; the code is valid for up to 12 months. This means that the $100 of Azure credit associated with the OSA must be used within 12 months. By default, when the amount of credit left reaches 30 percent of the initial purchase, you will be alerted via an email and through the Azure Management Portal. Additional credits can be purchased and applied to the account when you receive a notification—or at any other time you wish. It works a lot like my son's school lunch card. I deposit funds in a lunch account, and he buys lunches using his lunch card. I get reminders from the school when the account runs low, but I can add additional credit at any time. Details on the Azure Open Licensing can be found here:

```
http://azure.microsoft.com/en-us/offers/ms-azr-0111p/
```

Note that if a customer using Azure Open Licensing chooses to move to an Enterprise Agreement (EA), it is possible to move the unused Azure credit associated with Open Licensing to the EA by making a support request.

Enterprise Enrollments for Azure

For larger organizations wanting to adopt Azure, the idea of using a credit card to pay for monthly services is not ideal. Although it is possible to be invoiced by Microsoft, enterprises typically want more granular control of their expenditures. For example, they may wish to have high-level enterprise administrators who administer the entire Azure service for the organization. Those enterprise administrators can then create separate accounts and delegate account owners, who can group like services or groups together into subscriptions. The subscriptions can be used by service administrators and co-administrators. Azure Enterprise Enrollment allows exactly that. Organizations can add Azure services as part of their Microsoft Enterprise Agreements based on a certain amount of Azure spend. For example, an organization might make a commitment to spend $50,000 a year and with that agreement comes special pricing and possibly other benefits. There have been offers from Microsoft that a certain Azure commitment spend enables the organization to receive a free StorSimple storage appliance.

Beyond just the purchasing options, a key benefit for enterprise enrollment is the account and subscription flexibility and control. Typically, when an individual or small organization purchases Azure services, that user receives an individual Azure subscription with a specific subscription ID. That user is the service administrator for that subscription. Additional people can be made co-administrators. As of this writing, there is a limit of 200 or fewer co-administrators depending on the subscription type. Co-administrators can use the services within the subscription.

An enterprise enrollment via an Enterprise Agreement enables more flexibility in the administration and separation of services by providing three layers, as shown in Figure 1.12. Within the enterprise's enrollment, one or more accounts and subscriptions can be authorized.

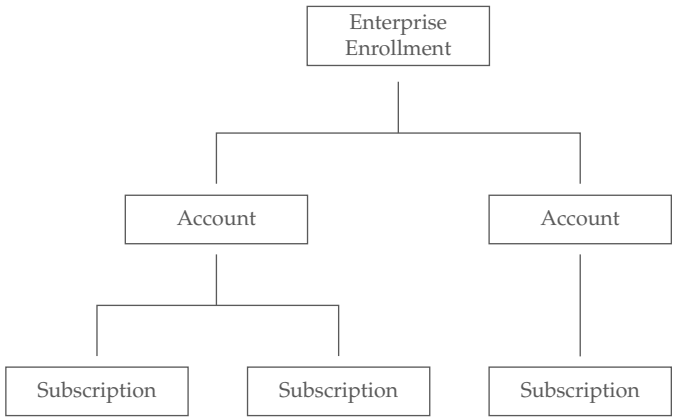


FIGURE 1.12
Hierarchy when using an enterprise enrollment

At the top of the enterprise enrollment is the enterprise administrator for the entire enterprise Azure enrollment. Nominate someone from your organization to receive an email and hold a Microsoft account. The work email address associated with the Microsoft account must be someone who will be able to activate the service. Once the service is activated, additional people from the organization can be made enterprise administrators. The enterprise administrators can in turn create separate accounts within the enrollment. Each account can have one or more account administrators assigned. Once again, each administrator needs a Microsoft account unless integration with your organization's AD has been implemented through the use of synchronization to Azure AD. If your AD and Azure AD are synched, organizational accounts in Azure can be used. (Active Directory is covered in detail in Chapter 7, "Extending AD to Azure and Azure AD.") The organizational account owners can create one or more subscriptions within the account. Each subscription has a single service administrator who manages the subscription and can add additional co-administrators who can use the services.

A key benefit is that the enterprise administrators at the top of the hierarchy have visibility into Azure usage across all accounts, whereas account owners have visibility into Azure usage for all subscriptions within the account. Billing reports are broken down at a subscription and account level for easy accounting. The hierarchical nature of an enterprise enrollment opens up a number of methodologies for setting up accounts and subscriptions, some of which are shown in Figure 1.13. Notice that in a functional methodology, the accounts are based on the functions of different groups. When using a geographical methodology, the accounts are based on physical locations. Business group methodologies often give each group their own account. Your organization may need a hybrid methodology. Ultimately, the right methodology depends on how you want to delegate the creation of subscriptions and how billing information and even chargebacks will be used within the organization. Once the account methodology is decided, how subscriptions are used must also be decided. This can be broken down by group,

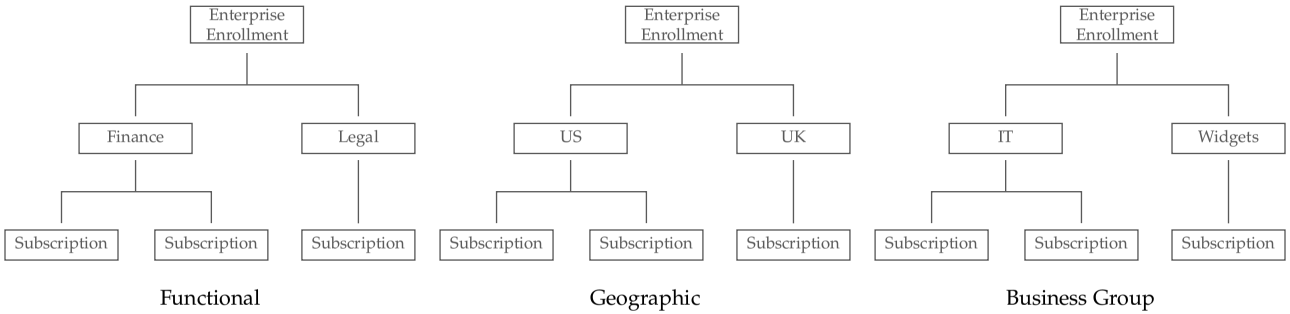


FIGURE 1.13
Possible methodologies for enterprise enrollment account setup

task, location, deployment life-cycle stage (such as a subscription for development and one for testing), and so on, ideally complementing the methodology picked for the accounts.

With the introduction of role-based access control (RBAC) and resource groups, some of the reasons for separate subscriptions and accounts no longer apply, since granular permissions, and even spend reporting, are now possible based on specific resources. But the concept of separating accounts and subscriptions can still be very useful.

Note that when paying by credit card or via invoice, you pay the list price for Azure services. However, when you purchase via an Enterprise Agreement, you can get substantial discounts depending on the level of the agreement. Based on the November 2013 Enterprise Spotlight report, the discount for EA agreements is based on the EA level, as shown in Table 1.2. The discount applies to the amount prepurchased under the agreement and to any overage charges beyond the amount in the agreement. This means no penalty for going over your prepurchase, and not surprisingly, Microsoft makes it pleasant to use more Azure! Depending on the amount of overage, you may be billed at the end of the year or potentially earlier; that is something you'd have to discuss with your Microsoft account representative.

TABLE 1.2: Enterprise-level Azure discounts

EA LEVEL	AZURE DISCOUNT
A	27%
B	30%
C	33%
D	36%

http://download.microsoft.com/documents/uk/licensing/resources/Spotlight_November_2013_to_Customer_-_AZURE.pdf

The discount available when purchasing Azure as part of an Enterprise Agreement is potentially huge, but it is important to realize the Azure purchase on an EA is prepurchased on an annual basis and is "use it or lose it." Suppose that as part of an EA you purchase $100,000 of Azure for Year 1. This gets you a certain level of discount. However, if at the end of the year you have only used $20,000 worth of Azure service, that other $80,000 is lost. It does not carry over to the next year. Customers can change the annual Azure amount each year, up or down, but they cannot reclaim any lost spend.

The risk of losing Azure dollars may encourage customers to commit to a lower up-front Azure spend, since any overage receives the same discount. That has to be balanced against the fact that the more Azure spend you commit to and pay, the greater the discount. Numerous calculators and tools are available to help you estimate the Azure spend and reach to as accurate a number as possible for the coming year. In general, I would err a little on the low side, since you get the same discount for overage anyway. Be aware that erring on the low side could place you in a different EA level band and therefore a lower discount. Also, remember that there are many services available in Azure that can be used to consume your bucket of Azure money. If you are not using as much IaaS as you planned, maybe you can use more storage or media services.

To summarize, Azure can be obtained in a number of ways, including:

- Trial subscription
- Pay-as-You-Go subscription
- MSDN Azure benefits
- Azure Open Licensing
- Enterprise Azure enrollment

Increasing Azure Limits

By default new Azure subscriptions are initially configured with very low limits. The default limits are in place to stop new Azure customers from initially over-consuming services. The limits can (and will need to be) increased for serious Azure usage. For most limits, there is a default value and a maximum value. Although it may be tempting to try to raise limits to the maximum, remember that the limits are there to help protect your own usage. I advise increasing to a realistic value that meets your needs. It's also critical to have proper processes in place to ensure that services are running only as needed and monitoring is in place to ensure that you have the required levels of insight to your environment. (I will cover those throughout this book.)

The Azure default limits and maximums are outlined here:

```
http://azure.microsoft.com/en-us/documentation/articles/azure-subscription-
service-limits/
```

The key limits you will want to change include the following:

Cores per Subscription Twenty cores per subscription is the default, which is a very low number and should be increased. Note that when using the A0 shared core instance size, the shared core counts as a whole core from a cores-per-subscription limit measurement.

DNS Servers per Subscription Nine DNS servers per subscription is the default. Many customers find this too low, especially when using Azure for testing and development and multiple virtual networks representing different environments.

Cloud Services and Storage Accounts per Subscription Both of these limits are set to 20 by default, but you may need to increase them based on how you build your Azure services.

To raise a limit, contact Azure support on the phone or via the Azure Management Portal. You'll find full details on how to get Azure support here:

```
http://azure.microsoft.com/en-us/support/faq/
```

The steps to complete an online quota request are as follows:

1. From the Azure portal, select your name from the top corner to display the hidden menu, and select the contact support option. (The exact text will vary depending on whether you are using the new or the old portal.)
2. For Support Type, select Billing and then click Create Ticket.
3. For Problem Type, select Quota or Core Increase Request.

4. For Category, select the quota you wish to increase, as shown in Figure 1.14.
5. Complete the required details, such as the desired new value, and then click Submit.

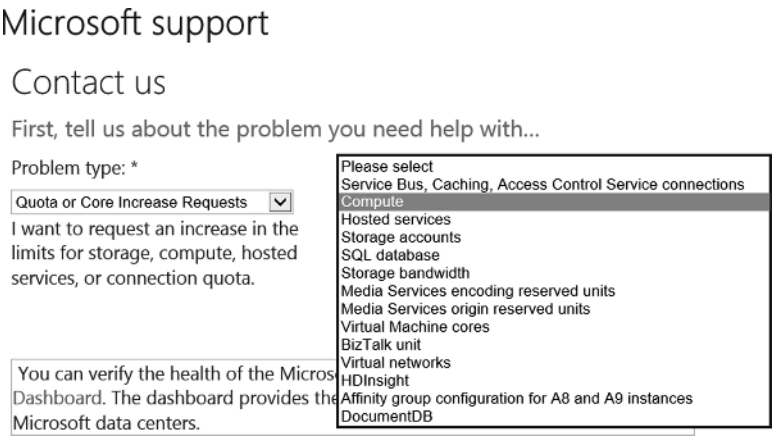


FIGURE 1.14 Selecting the type of quota to increase

The Bottom Line

Articulate the different types of "as-a-Service." Three core types of service are offered within clouds. Infrastructure as a Service (IaaS) provides a virtual machine with all the underlying compute, network, and storage fabric provided by the service vendor. The customer has full control of the operating system and services running within the virtual machine. Platform as a Service (PaaS) provides an environment where applications can be deployed without the customer having to consider the management or maintenance of operating systems or middleware services. Applications may be required to follow certain guidelines to run in a PaaS environment. Finally, Software as a Service (SaaS) provides a complete solution that requires customers to perform only basic administration. Good examples of SaaS are services such as Office 365.

> **Master It** Why do many organizations initially use IaaS but ultimately move to PaaS and SaaS?

Identify key scenarios where the public cloud provides the most optimal service. A key attribute of the public cloud is that customers pay only for what they consume. This means any type of scenario where the usage is not flat over time is a great fit for the public cloud; when there is a lull in activity, the customer can scale back the service and pay less. The public cloud can also be useful for start-ups that wish to avoid the initial outlay for hardware and software. Again, in the public cloud you pay only for what is needed.

> **Master It** How is the public cloud used in hybrid scenarios while still gaining the benefits of public cloud pay for consumption only?

Understand how to get started consuming Microsoft Azure services. For individuals who want to try Azure, the best way to gain access is to sign up for a free one-month trial. If you have an MSDN subscription, you can activate the Azure benefits for monthly Azure credits

to be used for test and development purposes. For organizations wishing to leverage Azure for production purposes, numerous options exist, including Pay-as-You-Go via credit card or invoice, Open Licensing, or adding Azure services to your Enterprise Agreement. A key point is to have a clear project that you wish to implement in Azure that will help you start leveraging Azure quickly.

Master It How can I avoid going over my MSDN monthly quota each month?