David Chappell

ADOPTING MICROSOFT AZURE

A GUIDE FOR IT LEADERS

Sponsored by Microsoft Corporation

Contents

Public Cloud Platforms: The Future of Enterprise Computing

There's no denying it: enterprise computing is moving to public cloud platforms. The benefits the cloud brings, including greater flexibility and lower costs, are becoming too obvious to resist. Starting now, every IT leader needs to think seriously about how they'll use cloud platforms.

But this is a big change. Relying on a cloud platform, such as Microsoft Azure, is different from what you're doing today. Making this transition will take time and thought. And the best place for you to begin isn't necessarily obvious. While you'll probably run mission-critical applications on a public cloud platform at some point, starting here isn't a realistic option for most organizations—it feels too risky. Yet there are other scenarios that can provide value immediately with manageable risk. You can start small, then expand when you're ready.

This overview describes how organizations can adopt Microsoft Azure. We'll look at the most common scenarios, starting with the simplest, and show the business value of each one. The goal is to help you think about which cloud scenarios might make sense for your organization today.

It's useful to group these scenarios into two broad categories:

- *Infrastructure scenarios* that improve the operational aspects of running an IT organization. The usual goal here is to provide the most reliable service at the lowest cost.
- *New application scenarios* that create custom software used by your employees or your customers. The overarching goal here is to create competitive advantage for your organization. You do this by developing and deploying new applications faster, exploiting technology innovation to differentiate your firm from your competitors, and in other ways.

As we'll see, a cloud platform can have value in both categories.

Addressing Cloud Concerns

Microsoft Azure provides computing services from datacenters in North America, South America, Europe, Asia, and Australia. These datacenters are large and highly automated, which is why they can offer scalable services and low prices.

But running your applications and storing your data in remote datacenters owned by somebody else can feel scary. This approach to computing also raises legitimate concerns about compliance with whatever laws and regulations your organization is subject to. Before walking through how you can use a cloud platform, it's important to look more closely at both of these issues.

Security

For just about everybody, the first concern that comes to mind with public cloud platforms is security. You wonder how you can rely on Azure or another cloud platform to handle this for you.

You might start by thinking honestly about whether your datacenter is more secure than an Azure datacenter. Microsoft very likely has more resources than you do to build and operate state-of-the-art security technologies, to

carefully vet the people who work in its datacenters, and more. If you think your datacenter is more secure than an Azure datacenter, you're probably mistaken.

Yet the real issue with using a public cloud platform isn't security—it's trust. If you're responsible for a datacenter today, you probably lie awake some nights worrying. Are your firewall settings correct? Are your administrators honest? You worry about these things because they're your responsibility, and so it's your job to get them right.

With a cloud platform, these issues are no longer yours to worry about. In fact, you have no control over them. If you tour an Azure datacenter, Microsoft will happily let you look at their servers and other hardware from a distance, but they won't let you examine the details. You can't check the firewall settings, for example, or look through the resumes of the datacenter administrators. Instead, you have to trust Microsoft to get this right. The reality is that using a cloud platform requires you to trust the platform's provider.

How do you build trust? Most often, you do it slowly. You start small, see the benefits, and then take the next step. This is exactly how organizations typically adopt Azure. If you're like most of your peers, this stepwise approach to the cloud is the route you'll take.

And ask yourself this: How much does using Azure require you to increase the trust you already have in Microsoft? After all, you regularly let Microsoft install whatever software the company chooses directly into your operating system via Windows Update. You probably don't worry about whether the next update will contain code that steals all of your data; you just trust Microsoft not to do this. Over time, Windows Update has provided real benefits, and you've learned not to worry about it.

Expect to walk a similar path with Azure. The truth is that the level of trust you already have in Microsoft—through Windows Update and more—is probably at least as great as what Azure asks of you.

Compliance

Once you've decided to trust a public cloud platform enough to get started, the next question that arises is often compliance. How can you be sure that it's legal for your organization to do this?

Answering this question can be challenging. Different industries have different requirements—financial services firms are typically more constrained than manufacturing companies, for example—and the rules also differ across countries. Add to this the fact that many of these laws and regulations were written before cloud computing existed, and the result is a complex stew of rules.

Still, the laws are being modernized, and the situation is getting clearer. It's obvious, for example, that using public cloud technology is acceptable in many situations. The huge growth in software as a service (SaaS) solutions such as Office 365 and Salesforce.com CRM makes this clear. Just as important, Azure has a range of third-party certifications that can make compliance easier. (For more on this, visit the Azure Trust Center.)

If you have concerns about whether you can move data to Azure and remain compliant, you might need to get advice from legal professionals. And there are some situations in some industries that probably won't be cloud-friendly for a while. But if you're like most organizations, you'll probably find that you can do more than you thought you could in the cloud while still complying with the necessary regulations.

Infrastructure Scenarios

Many IT leaders first use Azure to improve the operational aspects of IT—they start with infrastructure. The most common scenarios here include the following:

- Using cloud storage.
- Building a development and test environment.
- Providing single sign-on to SaaS applications.
- Supporting disaster recovery.
- Deploying packaged applications.
- Moving existing applications.

This section looks at each of these, describing both the scenario and why you might want to do it.

Using Cloud Storage

The simplest way to store data on Azure is to use Azure Blobs. A blob is just a collection of binary data, and so Azure blob storage acts much like a SAN in the cloud. Blobs can be used in many different ways: for backup data, for storing and streaming video, and more. Perhaps the most common way that organizations begin using Azure Blobs today, though, is through Microsoft's StorSimple appliance. Figure 1 shows how this looks.

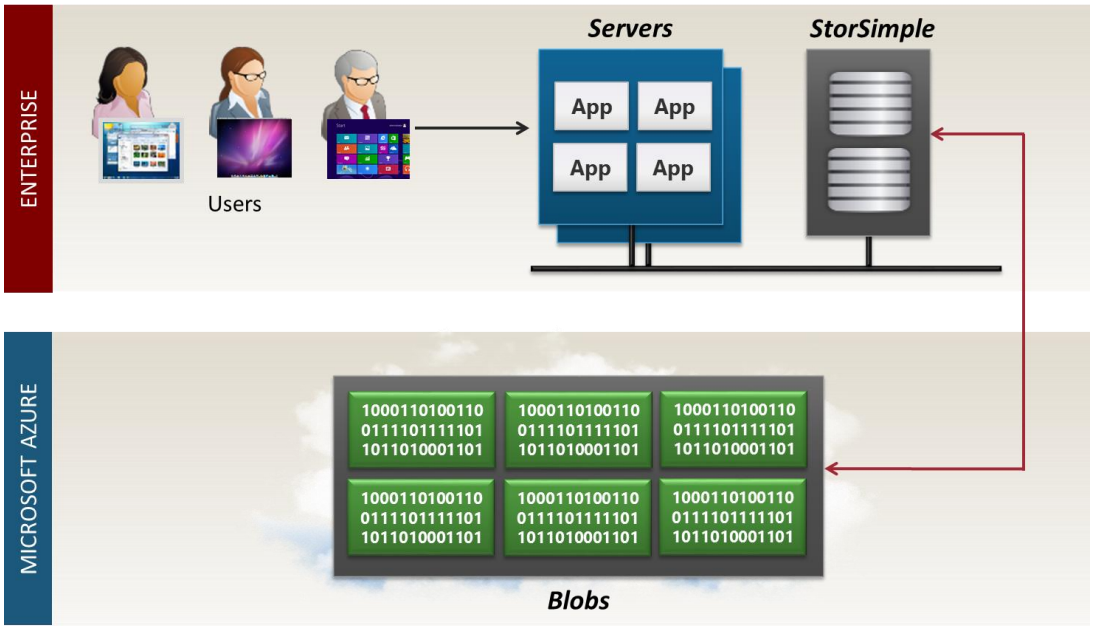

Figure 1: A StorSimple appliance moves data back and forth as needed between on-premises disk and lower-cost Azure Blobs.

As the figure illustrates, StorSimple provides hardware that sits in a customer datacenter. This device provides local storage—it has its own disk—but it also has a connection to Azure Blobs. Data is silently moved between the on-premises device and blobs based on how it's used. For instance, data that's accessed infrequently can be moved automatically to the cloud, then brought back on-premises when it's needed. The result looks like a large SAN, with a bit slower access to some of the data it contains.

Why do this? What's the benefit of using Azure Blobs? Most often, the answer is lower cost. Suppose, for example, that your organization stores one terabyte of information in a geo-redundant blob. (Geo-redundancy helps with disaster recovery by storing an instance of the blob in two Azure datacenters in the same region, such as the United States or Europe.) If your users consume 500 gigabytes of outbound bandwidth per month accessing this blob, the cost works out to around $105/month for Azure's US and Europe datacenters, $130/month for the Asia datacenters, and $150/month for South America[1]. If you're like most organizations, this is significantly less than the cost of a terabyte of on-premises storage. And because StorSimple encrypts the data it stores—leaving the keys with you, not Microsoft—it can be a lower-risk way to begin using this public cloud platform.

Building a Dev/Test Environment

Any organization that creates custom applications needs to provide a development and test environment for building those applications. A development team needs to install specific tools, while the test environment must replicate the world in which the new application will be deployed. Given the cost and time required to provision physical servers, it's become common to use virtual machines to do this.

One approach is to create these VMs on servers in an organization's own internal datacenter. Another option is to create them on Azure, as Figure 2 shows.

[1] All of the pricing numbers in this paper are current as of mid-2014. Prices change frequently, however, and they always go down. For details on current Azure pricing, see http://azure.microsoft.com/en-us/pricing/overview/.

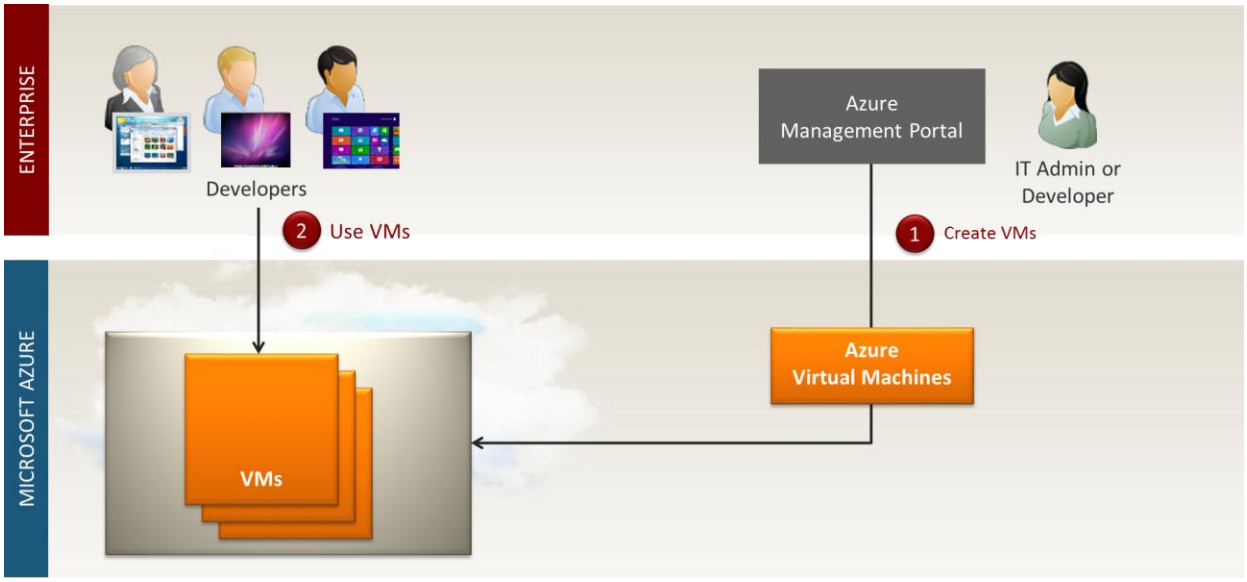

Figure 2: Azure Virtual Machines can provide an on-demand development and test environment for creating new applications.

As the figure shows, an IT administrator or a developer can use the Azure Management Portal to create VMs in the cloud (step 1). Those VMs are created using Azure Virtual Machines, the platform's Infrastructure as a Service (IaaS) offering. Developers can supply their own VM images or use images provided by Azure, with support for both Windows Server and Linux. Once the VMs exist, the development team can use them to build and test a new application, installing whatever software they need (step 2).

The main reasons that organizations take this approach are speed and low cost. Azure VMs are available to their users in a few minutes, while deploying VMs in an organization's own datacenter can take days or weeks. Also, an organization pays for public cloud VMs by the hour, at prices ranging from two cents to a little more than a dollar per hour. To make this approach even cheaper, you can shut down the VMs when they're not in use, such as at night and on weekends.

Using Azure for development and test is a low-risk way to begin using the cloud. The development process commonly uses test data that won't raise compliance issues, and developers are usually quite open to new approaches that improve their lives.

Providing Single Sign-On to SaaS Applications

Users hate remembering multiple usernames and passwords. They'd much rather log in just once, then be able to access all of their applications. In an on-premises Windows domain, this problem is solved by Active Directory, which gives users single sign-on (SSO) to the applications running in that domain.

As companies use more and more SaaS applications, however, this challenge reappears. How can you give your users SSO to a range of cloud software provided by diverse vendors? Azure Active Directory addresses this problem, as Figure 3 shows.

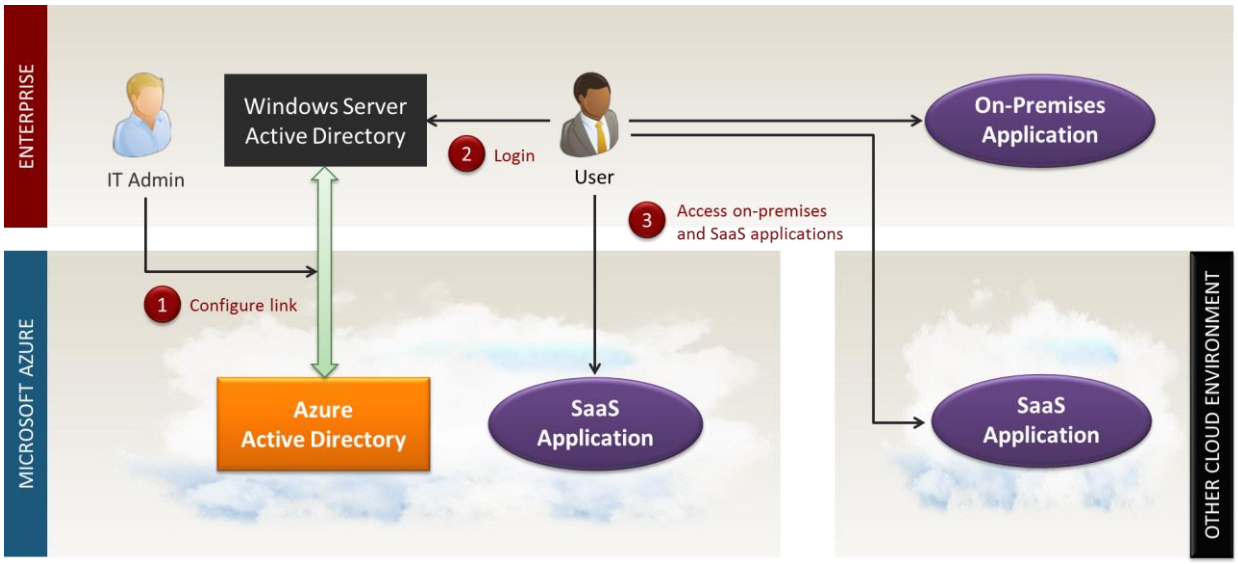

Figure 3: Azure Active Directory allows single sign-on to SaaS applications running on Azure or elsewhere.

To use this service, an IT administrator in your organization configures a link between your on-premises Active Directory running on Windows Server and Azure Active Directory (step 1). A user then logs into your Windows domain as usual (step 2), but he can now access both on-premises and cloud applications without signing in again (step 3).

As the figure shows, Azure Active Directory provides SSO both for SaaS applications running on Azure and SaaS applications running on other cloud environments. Today, for instance, Azure Active Directory supports Microsoft offerings such as Office 365 and Dynamics CRM Online. It also supports a range of SaaS applications provided by other vendors, including Google Apps, Salesforce.com CRM, Dropbox, and many more.

Just as Active Directory provides SSO for on-premises applications, Azure Active Directory solves this problem for the cloud era. And because at most only a hash of a user's password is stored in the cloud, this service can provide a useful but low-risk way to get started with a public cloud platform.

Supporting Disaster Recovery

Ensuring business continuity in the face of failure is a requirement for many IT leaders. The usual approach relies on maintaining backup hardware, sometimes in a separate datacenter, that can take over when something goes wrong. This approach works, but it's expensive; you're paying for hardware and software that's rarely used.

Why not use a cloud platform instead? Rather than buying and maintaining redundant physical systems, you can use Azure resources to keep an application running. In some cases, you can even pay for those resources only when a disaster occurs. Figure 4 shows a simple example of how an organization might use Azure to provide disaster recovery for an application.

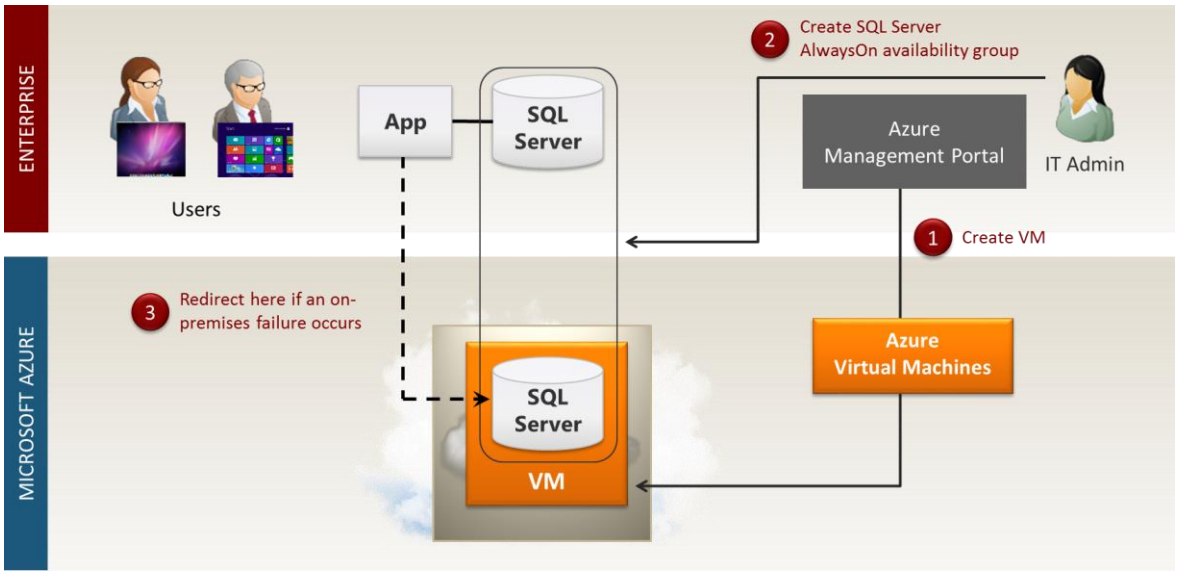

Figure 4: Azure VMs can be used to provide disaster recovery in the cloud.

In this scenario, the goal is to provide a backup instance of SQL Server for an on-premises application. To do this, an IT admin creates an Azure virtual machine running SQL Server (step 1). She then sets up a SQL Server AlwaysOn availability group that includes both the on-premises SQL Server instance and the cloud instance (step 2). Once she's done this, changes made to the on-premises database will be reflected in the cloud copy. If the on-premises instance fails, the application will automatically begin using the cloud version (step 3).

Azure also has other disaster recovery options. For example, Azure Site Recovery allows replicating on-premises VMs in the cloud. If the on-premises systems fail, the cloud versions can be started to take over the load. Rather than paying for backup hardware that sits unused most of the time, an organization can pay for Azure VMs only when they're needed.

Deploying Packaged Applications

A typical IT organization will start using Azure with one of the scenarios already described, such as using cloud storage or creating a development and test environment. Once you've built some trust, however, you can start to rely on Azure for more than this. For example, you might choose to deploy packaged applications in Azure VMs, as Figure 5 shows.

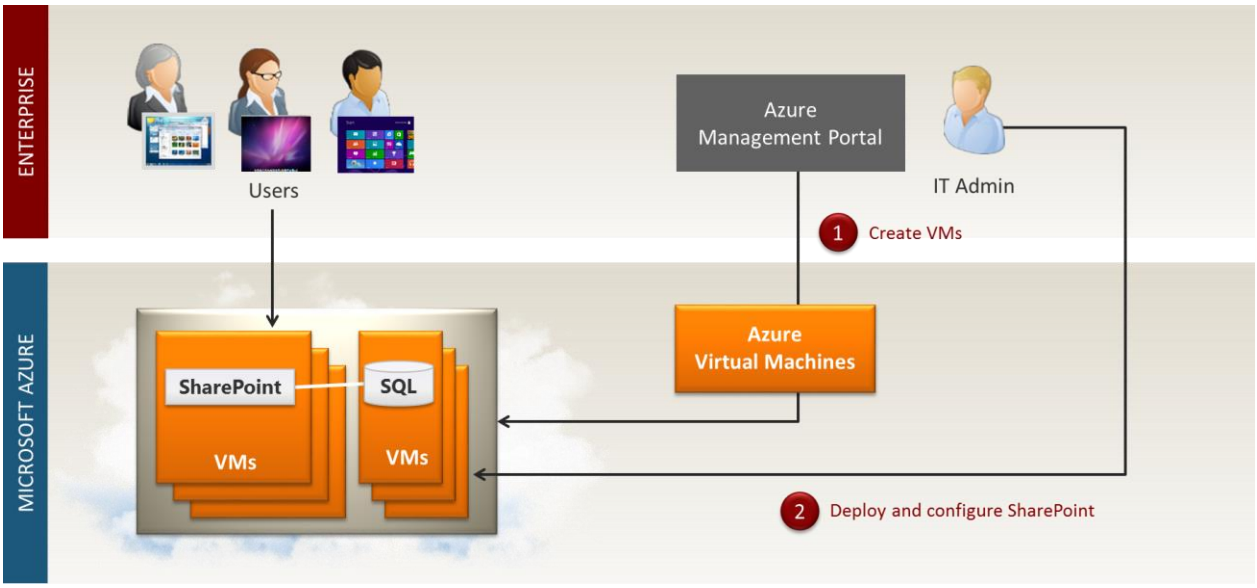

Figure 5: Azure Virtual Machines can be used to run packaged applications such as Microsoft SharePoint.

As in the previous scenarios, an IT administrator relies on the Azure Management Portal to create VMs using Azure Virtual Machines (step 1). The administrator then deploys and configures an application—in this example, it's SharePoint—in those VMs (step 2). Azure VMs support a variety of packaged software today, including Microsoft Dynamics applications, Oracle databases, and more.

Why run a packaged application on Azure rather than on servers in your own datacenter? One common reason is to allow faster deployment. Rather than wait for central IT to provide the necessary physical or virtual servers, a business unit can create Azure VMs in minutes, then immediately begin deploying the application. (And while some CIOs believe they can stop their business units from doing this, they're most likely mistaken.)

Running packaged applications in Azure VMs also lets IT resources become an operating expense rather than a capital expense. Organizations vary—some CFOs prefer this and some don't—but a public cloud platform makes the option available. Running packaged applications on Azure is also likely to be cheaper than running them on premises, especially as competition continues to force down cloud platform prices.

Be aware, though, that while Microsoft is responsible for ensuring the security of Azure datacenters, your organization is still responsible for application-level security. If you let your users choose weak passwords, for example, your application will have security challenges even if they run in an Azure datacenter.

Moving Existing Applications

If it's possible to deploy new packaged applications on Azure, then it's also possible to move existing applications to this public cloud platform. Sometimes called *lift and shift*, this approach once again relies on the IaaS support provided by Azure Virtual Machines. Figure 6 illustrates the idea.

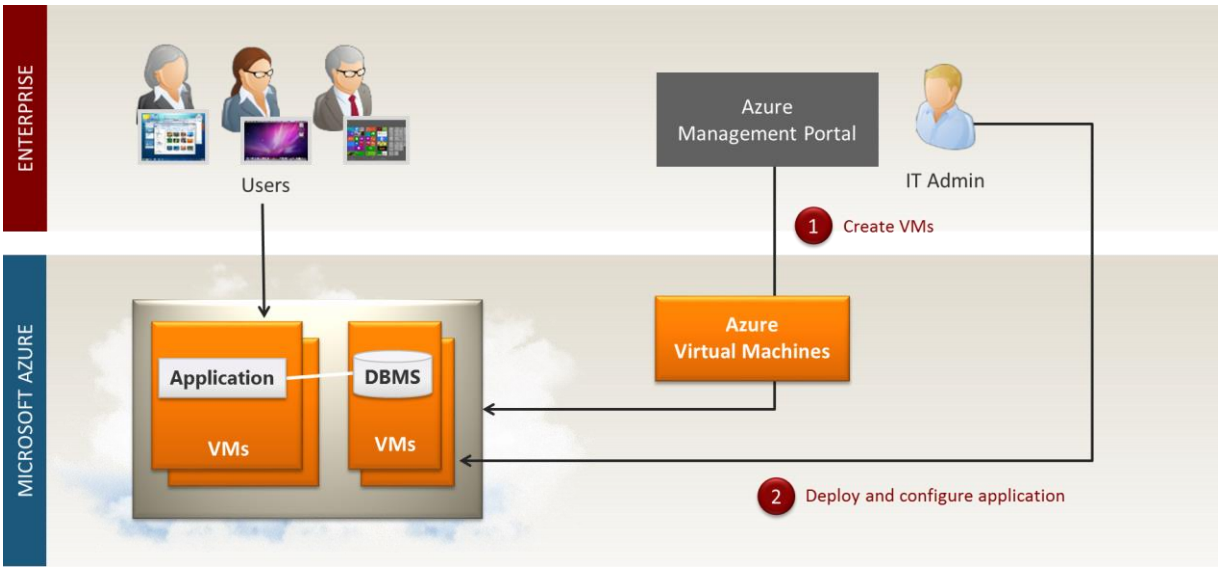

Figure 6: Enterprises can move existing applications to Azure Virtual Machines.

As before, the process begins with creating the VMs (step 1). Rather than deploying a new packaged application, however, the IT administrator instead deploys and configures an existing application to run in those VMs (step 2). It might be a .NET application using SQL Server, a Java application using Oracle, a PHP application using MySQL, or something else, and it might run on Windows Server or Linux. Azure supports all of these technologies.

Why do this? The obvious answer is the potential for lower costs. Suppose, for example, that an organization moves a relatively small application to Azure, one that can run in two medium Azure VMs. Suppose further that this application stores 100 gigabytes of data and uses 50 gigabytes of outbound bandwidth per month. With these assumptions, the total monthly cost of running the application in an Azure datacenter will be around $280 per month.

Is this cheaper than running the application on premises? Is it more expensive? The honest answer for most organizations is that they simply don't know; few IT departments track their costs on a per-application basis. This can make it hard to calculate the value of moving an existing application to Azure, which in turn makes this move hard to justify. If you're considering this scenario, the place to start is by working out your current costs for running an on-premises application. Only then can you really understand the financial arguments for moving those applications to Azure.

When you compare Azure costs to on-premises costs, though, be sure to do an apples-to-apples comparison. It's tempting to compare, say, just the cost of buying a server to the annual costs of Azure VMs and storage. This isn't correct, however, because using that server will incur other costs: power, cooling, datacenter space, people to administer it, and more. To make good decisions, you need to compare the fully burdened cost of a server with the cost of using the same resources on Azure.

New Application Scenarios

Rational organizations create new custom applications for only one reason: to do something unique. For a business, this generally means something that brings competitive advantage. Building those applications on a cloud platform can provide a number of benefits, including faster development and deployment, more flexibility in how resources are used, and lower cost.

It's useful to group new application scenarios for cloud platforms into three categories:

- Creating new employee-facing applications.
- Creating new customer-facing applications.
- Creating new parallel applications.

This section describes how and why you might use Azure in all three areas.

Creating Employee-Facing Applications

Using Azure to support new applications for an organization's own employees can look much like deploying packaged applications. Figure 7 shows one approach.

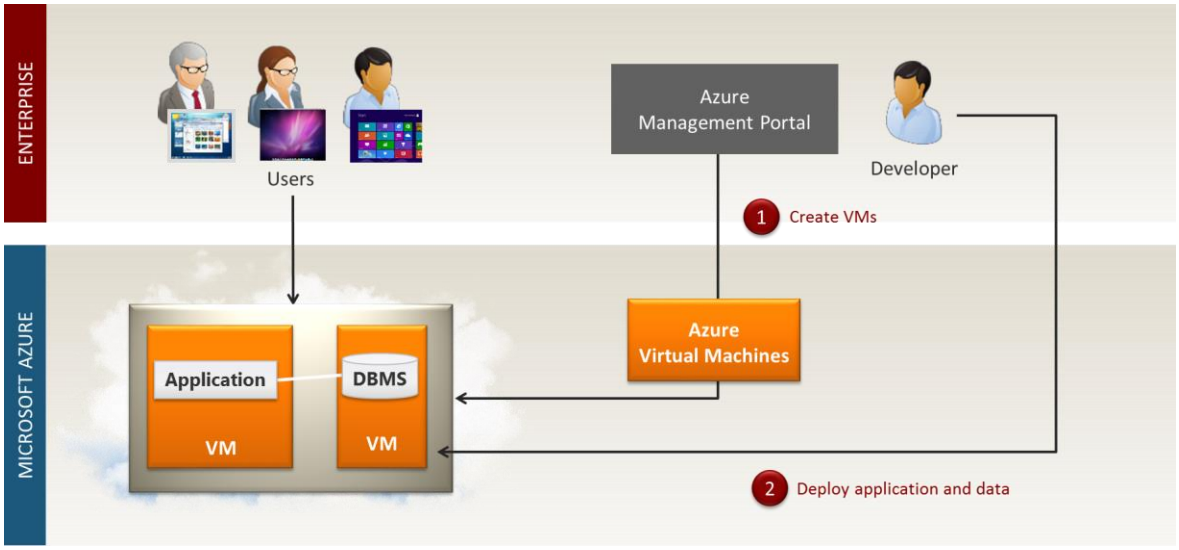

Figure 7: Organizations can create new employee-facing applications that run on Azure IaaS.

As in the scenarios shown earlier, an IT administrator creates VMs using Azure's IaaS technology, Virtual Machines (step 1). Once these are available, a newly created application and its data can be deployed and run in those VMs (step 2). It's likely that the development and testing of this application will also be done using Azure VMs, although it's not strictly required.

Why would an organization choose Azure as the foundation for a new employee-facing application? One possible reason is that this public cloud platform provides capabilities that might be hard to get otherwise. Suppose the

application needs to be accessible to employees around the world, for example. Because Azure has datacenters on many continents, the application can be deployed near employees wherever they might be, minimizing network delays.

A public cloud platform's usage-based pricing can also be beneficial with some kinds of employee-facing applications. Think about an application that's used only once a month, for instance, but must support many simultaneous users during these monthly peaks. An enterprise could run multiple instances of the application in many VMs during the peaks, then run only a couple of VMs in the slow times. Because you pay for Azure VMs by the hour, this can be less expensive than using computing resources in your own datacenter.

A business unit might also choose to build a new employee-facing application on Azure to go around its own central IT organization. Cloud platforms give anybody fast access to computing resources, regardless of what central IT might prefer. When a business unit is creating a new application for competitive advantage, such as improving an internal business process, it won't be happy waiting for VMs when a faster alternative is available—it wants the business value now.

Creating Customer-Facing Applications

Just as building new employee-facing applications on Azure can make sense, so can creating new applications used by your customers. Figure 8 shows how this might look.

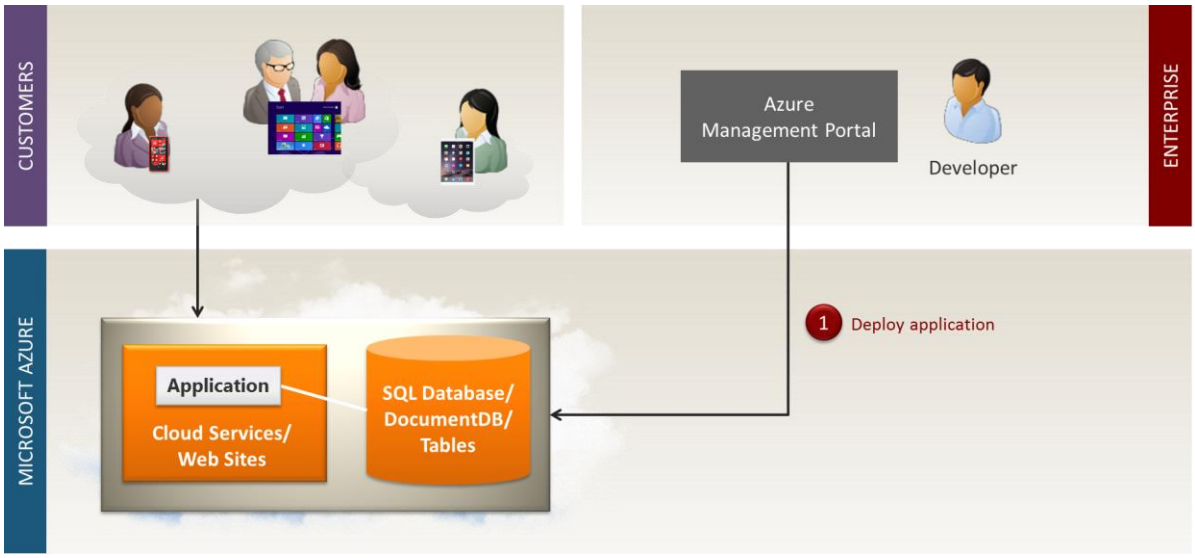

Figure 8: Enterprises can create new customer-facing applications that run on Azure's PaaS technologies.

This scenario is quite similar to the previous one, but with two differences. First, the application is now used by customers accessing it via the public Internet. Second, the application isn't built on Azure Virtual Machines, which provides Infrastructure as a Service. Instead, it's built using Azure's Platform as a Service (PaaS) technologies. Unlike IaaS, which provides ordinary virtual machines, PaaS offers a managed platform for running applications and working with data.

With PaaS, there's no need to first create VMs, then deploy an application into them. Instead, a developer just deploys the application and its data (step 1). Users see no difference—they access the application as usual.

Azure provides several PaaS technologies today. For running applications, developers can choose either Cloud Services or Web Sites. For working with data, the options include SQL Database for relational data, along with DocumentDB and Tables, both of which provide NoSQL data stores.

While it's certainly possible to create new Azure applications using IaaS, PaaS is often a better choice. Since there's no need to explicitly create VMs, using PaaS makes development and deployment faster. Because the platform is managed by Azure, using PaaS saves on management costs. And because developers have fewer things to configure, PaaS can be less risky, since there are fewer chances to make mistakes. PaaS also brings some restrictions, such as constraints on installing packaged software, but the advantages frequently outweigh the drawbacks. Whether you're creating a customer-facing or an employee-facing application, building it on Azure's PaaS technologies might be your best option.

But whether you choose PaaS or IaaS, why would your organization choose to create a new customer-facing application on Azure at all? Most often, the motivation comes from capabilities that a public cloud platform provides that aren't easy to get from other solutions. For example, the geographic distribution of datacenters can be important, as with employee-facing applications, since your new application might be accessed by customers in different locations. Another common reason for building a new customer-facing application on a public cloud platform is scale. Handling very large numbers of users is rarely required for applications used by an organization's own employees—most companies just aren't that big. But a successful consumer application can have tens or hundreds of thousands of simultaneous users, with big spikes in usage. Because a public cloud platform provides massive scale and pay-as-you-go elasticity, it helps with both of these things. And since the same architecture that makes an application scalable also makes it more reliable, an application built on Azure can provide better service to its users, something that has real business value.

Another reason that enterprises create customer-facing applications on Azure is to get computing resources quickly with no long-term commitment. Suppose, for example, that your organization needs to create a website for a marketing campaign. Rather than requesting VMs in your own datacenter, which probably takes a while and requires you to make some commitment, you can instead get computing resources from Azure. Those resources are available almost immediately, and you can shut them down—and stop paying—whenever you like. If the campaign is a big success, getting more computing resources to handle the increased load is fast and simple.

One more common example of where a public cloud platform makes sense is for server logic that supports mobile applications running on phones and tablets. These are often consumer apps with broad variations in usage, and so the scalability and elasticity that Azure provides are useful. And while it's possible to run the server logic for a mobile app in your own datacenter, most enterprises aren't set up to handle this kind of variable load in a cost-effective way. Just as important, enterprises are often reluctant to let consumer applications directly access services running in their own datacenters. It might be more secure to run the application's server logic on Azure, then move the necessary data between the enterprise datacenter and Azure datacenters as needed. Azure also offers a technology called Mobile Services that provides specific things needed by mobile applications, such as authentication and notifications.

Creating Parallel Applications

Along with employee-facing and customer-facing applications, Azure can also help organizations create parallel applications more effectively. In a parallel application, a user creates a number of virtual machines, then uses them all at once—in parallel—to solve a problem.

It's useful to divide parallel applications into two categories: high-performance computing (HPC) and big data. In an HPC application, the goal is to perform lots of computing in a reasonable amount of time. Problems such as simulating a car crash or estimating the long-term performance of a bond portfolio require this kind of compute-intensive processing. Breaking an application into independent components, then running those components simultaneously can speed things up considerably. Figure 9 shows how this looks on Azure.

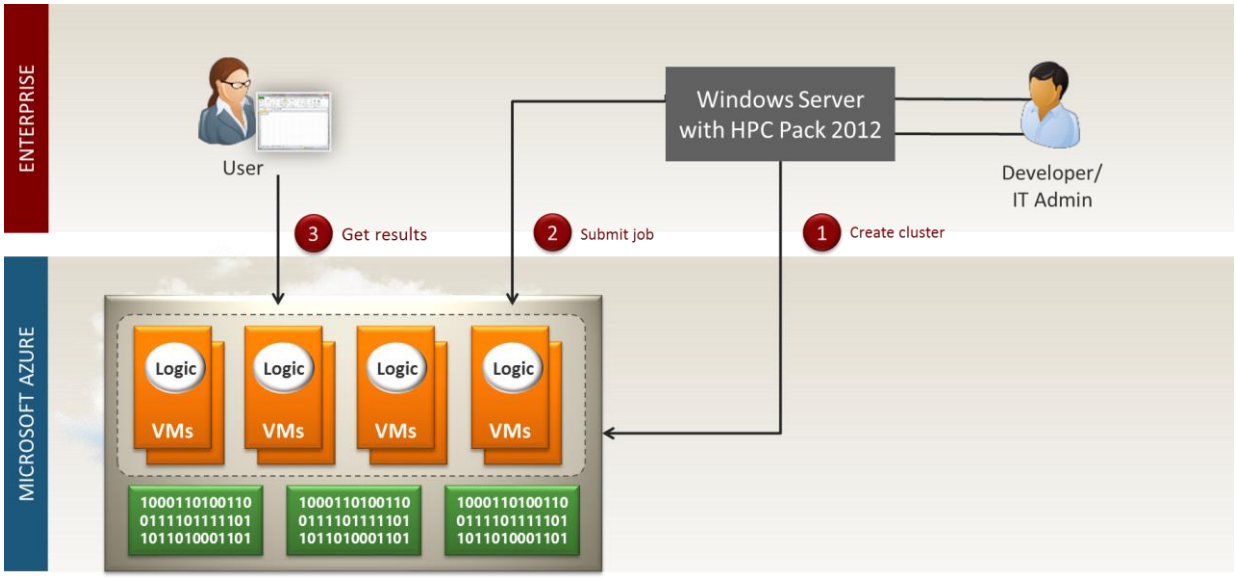

Figure 9: Windows Server with HPC Pack 2012 can create and manage compute clusters on Azure.

With Azure, an administrator can use Windows Server with HPC Pack 2012 to create a cluster of virtual machines (step 1). Once this has been done, the administrator or a developer submits an application (called a *job* in the HPC world) that runs on that cluster (step 2). A user can then access the results (step 3).

The advantage of running HPC jobs on a public cloud platform isn't hard to see. Rather than buying and managing a cluster of physical servers in an on-premises datacenter, an organization can instead spin up a cluster of Azure VMs on demand to run a parallel application. This can be significantly cheaper, which means it lowers the cost of entry to HPC. Simulations and other computationally intensive tasks that could once be performed only by companies able to afford their own dedicated cluster are now available to small and mid-sized organizations. And because Windows Server with HPC Pack 2012 has built-in support for creating and managing Azure clusters, doing this doesn't require extraordinary skills.

The second category of parallel applications, big data, is similar in some ways. As with HPC, it relies on using a cluster of VMs, with logic running simultaneously in all of them. With big data problems, however, the goal of using a cluster isn't to get lots of CPU power at once. Instead, the application needs lots of simultaneous data access.

For addressing this kind of big data problem, our industry has largely converged on a single technology: Hadoop. Azure provides a managed Hadoop service called HDInsight, as Figure 10 shows.

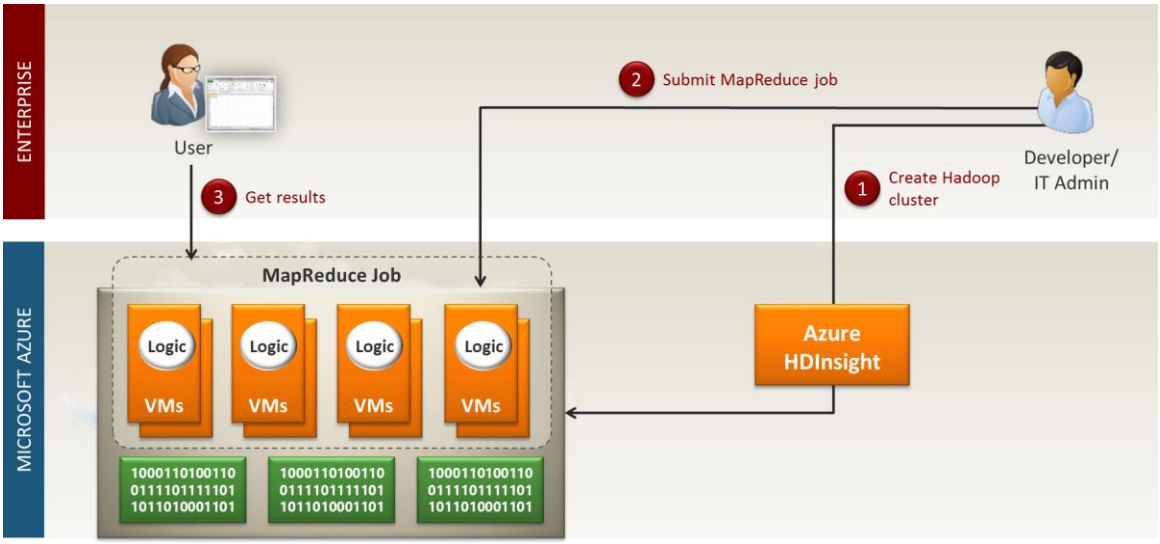

Figure 10: Azure HDInsight provides a Hadoop service.

An IT administrator or a developer can use HDInsight to create a Hadoop cluster (step 1). Once the cluster exists, a developer can submit a job that runs on this cluster (step 2). Hadoop jobs today typically use a programming model called MapReduce, although HDInsight also provides tools that let developers work at a higher level. Once the job completes, its results can be accessed by users via Excel or other software.

The motivation for working with big data on Azure is similar to the reason that HPC makes sense in this environment. Rather than buying and maintaining your own cluster of physical servers, you can instead create a cluster on demand, paying only for what you use. And since creating and managing a Hadoop cluster isn't simple, letting HDInsight do it for you makes sense.

Final Thoughts

Today, most of the computing and storage resources that your organization uses are probably in your own datacenter. Over time, however, expect this functionality to migrate to public cloud platforms. They'll be cheaper, and outsourcing computing infrastructure will let your organization focus less on running IT and more on getting the benefits that IT provides.

To reach this destination, though, you need to get started with public cloud platforms today. Whether you're focused on infrastructure scenarios, such as lowering storage costs, or new application scenarios, such as building a new customer-facing service, using a public cloud platform probably makes sense for you right now.

What are you waiting for?

About the Author

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.