Microsoft

Introducing Azure Stack

What It Is and When to Use It

Contents

A Holistic View of Cloud Computing

For most people, the word “cloud” conjures up an image of massive datacenters run by global providers that are accessible via the internet. And this image isn’t wrong—it’s just incomplete. There’s more to cloud computing than public clouds.

Perhaps the best definition of cloud computing was created several years ago by the US National Institute of Standards and Technology (NIST). In this document, NIST states that cloud computing has a set of essential characteristics, including *on-demand self-service*, *resource pooling*, *rapid elasticity*, and *measured service*. The NIST definition also specifies several possible cloud deployment models, including *public cloud*, *private cloud*, and a combination of the two called *hybrid cloud*. As this definition makes clear, cloud isn’t just a place—it’s a model that can be applied in multiple ways.

Microsoft Azure has provided public cloud services since 2010. Microsoft has also offered private cloud technologies with Windows Server Hyper-V, System Center, and Azure Pack. Yet because these public and private cloud services were created using different technologies, they weren’t as consistent as they might have been.

Microsoft Azure Stack changes this. This new offering provides a subset of Azure public cloud services as software that can be deployed in enterprise datacenters or in datacenters run by service providers (i.e., hosters). Figure 1 illustrates this idea.

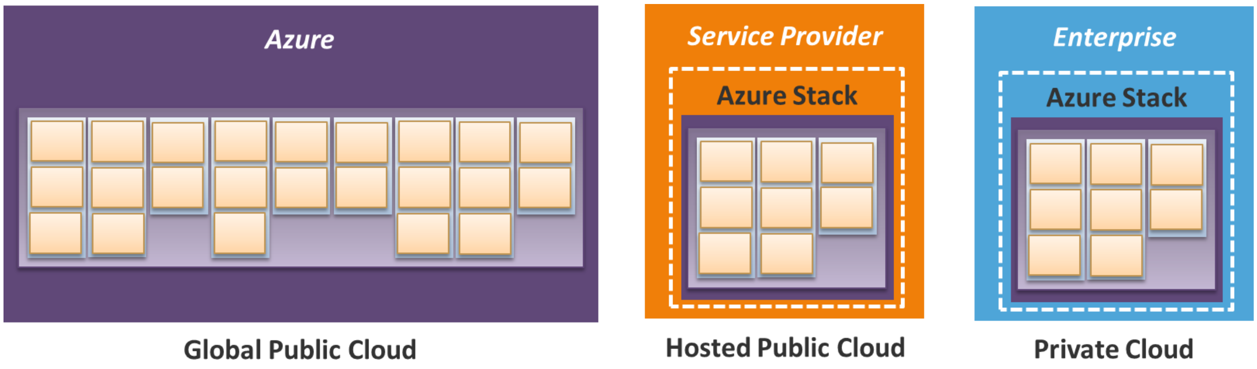

Figure 1: Azure Stack provides a subset of Azure technologies as software that can run in service providers and enterprises.

Azure Stack makes Azure services available in a consistent way across the global public cloud provided by Microsoft, hosted public clouds offered by service providers, and private clouds within enterprises.

Providing cloud technology in this way lets on-premises environments and service providers fully participate in the innovation and agility that Azure cloud technologies bring. It also has a number of other benefits, including the following:

- *Application developers can be more productive.* Azure Stack brings Microsoft’s most advanced development platform into your organization, with ongoing updates. Because your developers have access to these modern technologies, they can build better applications more quickly.

- *Applications and data can be moved between Azure Stack and Azure unchanged for both Windows Server and Linux.* The APIs are the same in both cloud technologies. And because Azure Stack includes Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) options, developers can create applications using both approaches without worrying that those applications will run only in the public cloud or in their private cloud. With Azure and Azure Stack, both options are available. This also means that organizations can easily rebalance their workloads across public and private clouds as their requirements change.

- *Managing your infrastructure gets simpler.* Microsoft has run a public cloud for many years, and we've learned a great deal about how to build and manage cloud infrastructure. Azure Stack brings this expertise to enterprises and service providers, letting you benefit from our experience.

- *Knowledge of Azure technology is relevant across both private and public cloud.* This skill set portability makes it easier to deploy people where they're most needed; you don't need to maintain groups of people with different abilities for different kinds of clouds.

- *Because Azure Stack provides a turnkey solution, enterprises can create a private cloud without the complexity of older approaches.* This private cloud can then be linked with Azure to create a hybrid cloud, or it can remain separate. In either case, enterprises get leading edge cloud services while still keeping all of their code and data inside their own firewalls, something that's often required due to regulations, concerns about data sovereignty, and other issues.

- *Internal IT organizations can provide the cloud services their developers demand.* Rather than adopting the public cloud, with the security and regulatory challenges this can bring, developers instead get these services in house.

- *Service providers can offer leading edge cloud services anywhere in the world.* Rather than create their own cloud offering or struggle with complex open source options, a service provider can take advantage of Microsoft's large and continuing investment in cloud technology (including documentation and training). It can also deploy this leading-edge cloud offering in any geography. For example, suppose a city government wants to use modern cloud technology, but is constrained by law to use only hosters in their immediate geographic area. A service provider can meet this need by offering a local datacenter running Azure Stack.

Some cloud vendors today provide only a public cloud service, while others focus largely on private cloud software. With Azure and Azure Stack, Microsoft uniquely offers a unified technology across public, private, and hybrid clouds.

Understanding Azure Stack

To understand Azure Stack, you need to know some basics about the cloud technologies this software provides. You also need to know a bit about how Azure Stack is deployed and managed within an organization. This section looks at these topics.

Technologies

Azure, the public cloud, offers dozens of services. Over time, Azure Stack will provide more and more of these. In its initial release, however, Azure Stack includes the subset of services shown in Figure 2.

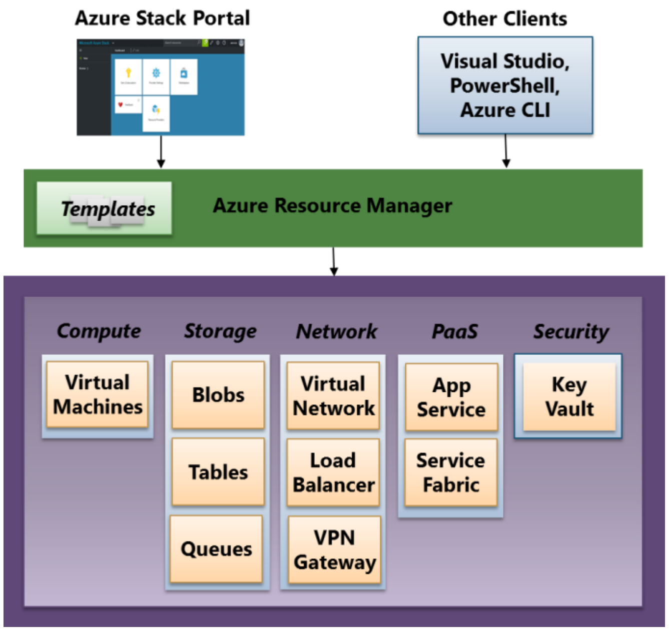

Figure 2: Azure Stack includes services for compute, storage, networking, PaaS applications, and more, along with management tools.

The technologies provided in the first release of Azure Stack include the following:

- *Compute:* Provides Azure Virtual Machines (offering Windows and Linux VMs on demand) and VM Extensions (allowing VM customization).
- *Storage:* Provides Blobs (for storage of unstructured data), Tables (a NoSQL key/value store), and Queues (letting cloud software communicate via messages).
- *Networking:* Provides Virtual Networks (allowing the creation of isolated networks in the cloud), Load Balancers (for spreading requests across replicated instances of an application), and VPN Gateway (allowing connections among virtual networks and more).
- *PaaS application platforms:* Provides App Service (supporting Web Apps, Mobile Apps, and API Apps created using .NET, Java, PHP, or other technologies) and Service Fabric[1] (providing a foundation for microservices applications).
- *Security:* Provides Key Vault (for securely storing encryption keys).
- *Automated deployment and more:* Provides Azure Resource Manager (exposes RESTful APIs to Azure Stack services and allows creating templates to automate deployment of Azure resources).
- *Management:* Includes Azure Stack Portal (providing a graphical tool for working with Azure Stack services) and support via Azure Resource Manager for other clients (Visual Studio, PowerShell, and a command-line interface (CLI) for Linux, Macintosh, and Windows).

[1] In preview at Azure Stack's initial release.

Deployment

In Azure, computing resources are organized into *regions*. There's a West US region in California, for example, a North Europe region in Ireland, and dozens more. When you deploy Azure Stack, your computing resources are also grouped into regions, as Figure 3 shows.

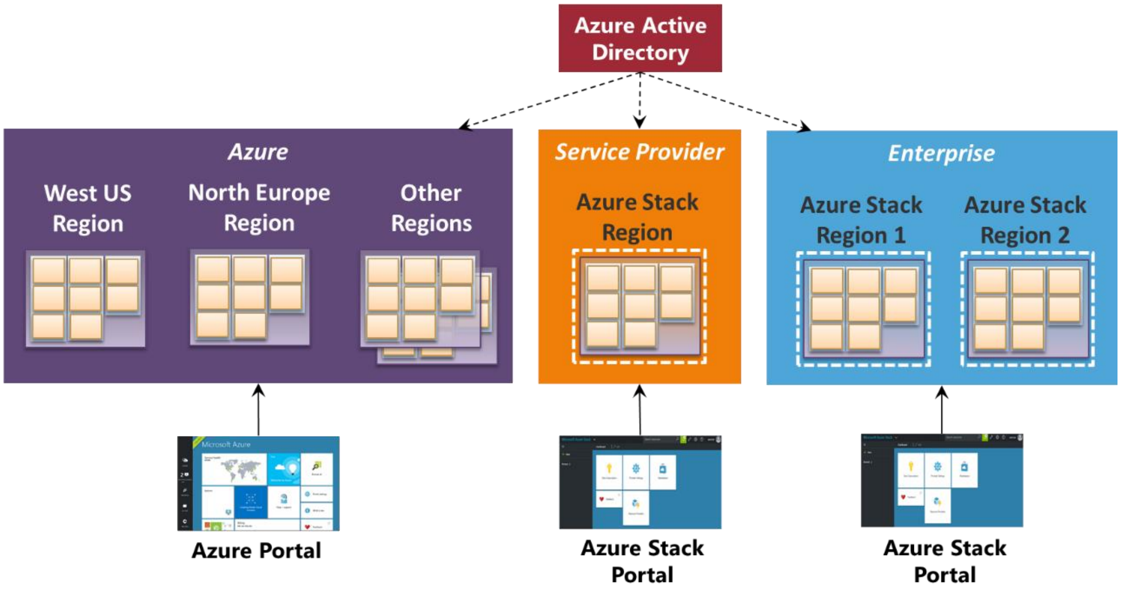

Figure 3: Like Azure, Azure Stack groups computing resources into regions that can be accessed via a portal.

An organization deploying Azure Stack might have just a single region, like the service provider in Figure 3, or it might have multiple regions, like the enterprise shown in the figure. You're free to structure your deployment in the way that best fits your needs. While the initial Azure Stack release will have some constraints, regions are designed for large-scale use. For instance, a single region can potentially include multiple datacenters, with Azure Stack hiding the seams between them.

As Figure 3 shows, access to Azure is through web-based portals. The Azure portal provides access to the global public cloud, while the Azure Stack portal is used to access clouds created using Azure Stack. These two portals share the same code, and provide much the same functionality. For example, each one lets its user specify the region she wishes to work with. This strong similarity makes it easy for people to work across both Azure and Azure Stack clouds; the portals can just be different tabs in their browser.

It's possible to connect Azure Stack clouds to Azure Active Directory as Figure 3 shows. This lets an organization use the same identities with Azure Stack as it does with Azure, Office 365, and other cloud services. Organizations can also use a local Active Directory service to serve their identity and access needs with Azure Stack – this is made possible through Active Directory Federation Services (ADFS) integration.

Cloud Computing is More than Virtualization

Virtualization provides real value. But cloud computing is much more than just virtual machines.

For one thing, cloud computing embodies all of the characteristics specified by NIST. It provides on-demand self-service access to resources, including VMs and more. A cloud provides pools of these resources, letting them be shared intelligently (and safely) across different users, along with an easy way to increase and decrease the number of resources you're using. A cloud also lets you pay for resources as you use them, an option that can lower your fixed costs considerably.

Just as important, while virtualization focuses on *servers*, cloud computing focuses on *services*. These services make application development faster and easier, provide broader options for working with data, and enhance your computing environment in other ways. This implies that you can't make the move to cloud computing just by improving how you do virtualization. The two are fundamentally different, and so getting to cloud computing requires a different approach.

Both cloud computing and virtualization have value, and the two will live side by side for quite a while. Still, while virtualization is useful, cloud computing offers much more.

Management

Azure Stack runs on premises in the organizations that deploy it. Because of this, Azure Stack clouds are managed by you (or perhaps a systems integrator working with you); Microsoft doesn't manage them remotely. For example, Microsoft supplies updates, just as with other software, but it's up to you to decide when to apply those updates. (You should expect them to arrive more frequently, though, reflecting the faster updates times typical for public-cloud Azure.)

Yet managing Azure Stack is different in important ways from what you do now. For example, you can't put off installing updates indefinitely—they must be applied within a specific amount of time to ensure Azure-consistency. More important, Microsoft is quite prescriptive about what hardware Azure Stack runs on. To get a sense of what this means, it's useful to take a closer look at what's inside an Azure Stack region. Figure 4 illustrates this.

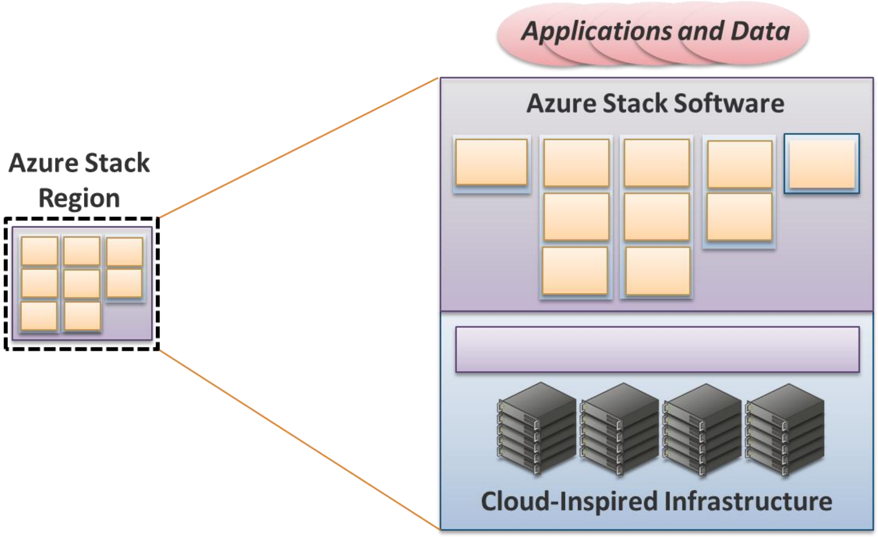

Figure 4: Azure Stack software runs only on hardware configurations specified by Microsoft.

As the figure shows, each Azure Stack region contains cloud-inspired infrastructure, consisting of racks with compute, storage, and networking, along with specialized software for supporting Azure Stack. This software isn't sold on its own, however. Instead, you purchase pre-built systems that include both Azure Stack software and hardware from Microsoft partners such as HP Enterprise, Dell, and Lenovo. The goal is to significantly simplify the process of creating your own cloud, helping get these services to application developers as quickly as possible.

Earlier on-premises cloud solutions often failed because of their complexity; there were too many moving parts. Microsoft has learned from this experience, and so with Azure Stack, we're providing something close to a turnkey solution. Because Microsoft has tested the approved software/hardware combinations, you can be confident that they work well together.

In fact, because all of these components are so integrated, you don't need to worry about tweaking the hypervisor or the storage system or many other things—Azure Stack optimizes these things for you. The simpler, more reliable approach that Azure Stack brings presents an opportunity to evolve your organizational model. Instead of spending as much time deploying and operating infrastructure, your IT staff can build proficiencies in deploying and operating cloud services.

The key thing to realize is this: Azure Stack changes how you manage your local environment. We're bringing real cloud technology—hardware and software—to your datacenter, which requires you to think differently. Don't expect Azure Stack to look like what you've seen before.

Using Azure Stack

Along with defining public, private, and hybrid cloud, the NIST definition also calls out the distinction between IaaS, PaaS, and Software as a Service (SaaS). Azure addresses all of these, and since Azure Stack brings Azure into your datacenters, it shouldn't be surprising that it also supports all three. This section looks at how Azure Stack can be used with each one of these options.

Whatever approach you're using, it's important to understand that Azure Stack draws a sharp line between the cloud provider and cloud consumers. Figure 5 shows how this looks.

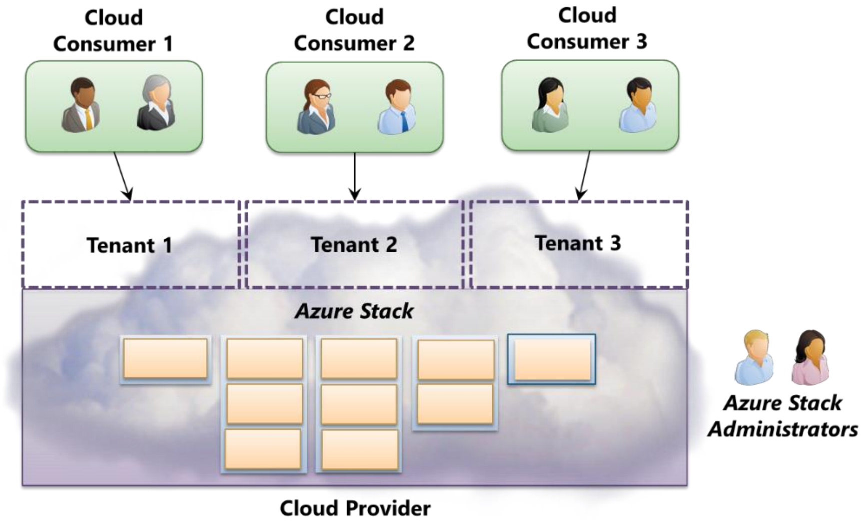

Figure 5: An Azure Stack cloud provider offers a multi-tenant service to its consumers.

The services Azure Stack provides are multi-tenant, which means that many consumers can simultaneously use the same service while maintaining isolation from each other. If the cloud provider is an enterprise IT organization, the cloud consumers might be different departments within that enterprise. If the cloud provider is a hoster, the cloud consumers might be different customers of this hoster.

Whether the cloud provider is an enterprise IT organization or a service provider, its administrators must define some number of *plans*. For example, a plan might include Azure Stack's compute, storage, and networking services. Plans are then grouped into *offers*, which are what cloud consumers actually subscribe to, with per-tenant billing. Offers are what define the strict boundary between an Azure Stack cloud provider and the cloud's consumers.

Offering IaaS Services

According to NIST, IaaS is the ability "to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software". Even though this is a fairly basic service, it's a fundamental part of cloud computing, and so it's part of both Azure and Azure Stack.

An Example

IaaS includes a range of different services. Figure 6 shows some examples of how Azure Stack can support this style of cloud computing.

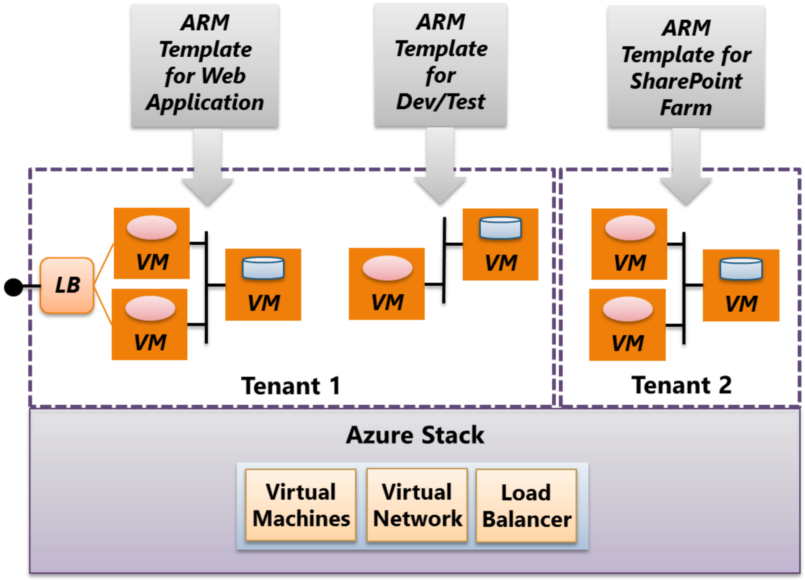
ARM Template for Web Application
ARM Template for Dev/Test
ARM Template for SharePoint Farm
LB
VM
VM
VM
VM
VM
VM
VM
VM
Tenant 1
Tenant 2
Azure Stack
Virtual Machines
Virtual Network
Load Balancer

Figure 6: Azure Stack supports IaaS, with ARM templates to automate deployment.

This scenario shows two tenants, each using the same Azure Stack services. They use these services to do different things, however.

- Tenant 1 has created a web application whose logic is replicated across two IaaS VMs. These VMs rely on a software load balancer that exposes a public IP address, then distributes user requests between them. For storage, the application uses a traditional database system running in a VM. All of these VMs communicate using a virtual network. Tenant 1 is also running a separate, simpler environment for developing and testing this application. Because it runs on a different virtual network, this dev/test environment is isolated from the production application.

- Tenant 2 has created a SharePoint farm. This farm runs in three VMs, all connected by a virtual network.

Because they run as different tenants, neither cloud consumer can access any of the resources being used by the other. Azure Stack ensures that these are kept entirely separate.

To define the isolated virtual networks shown in Figure 6, an Azure Stack consumer can use just the Azure Stack portal—there's no need to configure any hardware. Similarly, setting up the load balancer can be done entirely through the portal, with a completely point-and-click experience. By providing this kind of software-defined networking, Azure Stack makes creating and changing network configurations easier than with traditional approaches.

The Role of Azure Resource Manager

As Figure 6 suggests, Azure Resource Manager (ARM) plays an important role in Azure Stack. ARM provides a consistent approach to exposing the RESTful APIs that different Azure and Azure Stack services offer. It also, as the figure shows, allows you to create deployment templates.

Using ARM, cloud consumers, such as developers, people who deploy SharePoint farms, and others, can define *resource groups*. Each resource group specifies a particular set of Azure resources that should be deployed. For example, the tenant 1 ARM template shown in the upper left of Figure 6 defines a resource group that contains the three VMs this application requires, along with the virtual network they rely on. It also specifies the load balancer, the application code, the database system, and perhaps more. The next ARM template in the figure defines a resource group containing two VMs and a virtual network—it's a simple dev/test environment—while the ARM template used by tenant 2 defines a resource group that includes everything required to run this SharePoint farm. To deploy any of these, a cloud consumer simply asks ARM to execute the correct template. ARM takes care of everything else.

You're free to create your own ARM templates as needed. Visual Studio provides tools for doing this, and ARM SDKs are available for many different languages. ARM also has a command line interface (CLI) that runs on Windows, Linux, and Macintosh. And since lots of organizations need to do similar things, many pre-defined templates are available on github. Created by Microsoft and the larger community of ARM users, these templates address a variety of common situations, making it easier to get started.

ARM lets cloud consumers automate the mechanics of deployment. And since it works identically on Azure Stack and Azure, it's easy to move workloads—and skills—between these two clouds. Also, as described later, ARM can be used with PaaS and SaaS as well—it's not just for IaaS scenarios. This ability to automate on-premises cloud deployment, from simple dev/test environments to high-availability SharePoint farms and more, then easily move those deployments across Azure clouds, is one of most useful aspects of Azure Stack.

Why Do This?

For a service provider today, providing IaaS to its customers is table stakes—it's essential for offering a cloud service. Using Azure Stack for this, rather than a home-grown alternative or an earlier cloud technology, provides the benefits listed earlier, including a way to offer leading edge cloud services in any geography, compatibility with a major public cloud, and easier infrastructure management. With ARM, it also gives customers a simple way to move workloads from Azure or from private clouds running Azure Stack.

Even though Azure Stack is available to any service provider, hosters offering Azure Stack services can still differentiate from their competitors (and from Microsoft). One service provider might offer better support than its peers, for example, or promote its ability to provide more personalized service than Microsoft, or perhaps focus on meeting the compliance requirements for a particular industry. And while a hoster's customers will likely use the Azure Stack portal for many things, each hoster is also free to differentiate in other areas with its own portal. And through the Cloud Service Provider (CSP) program, Microsoft makes it easy for hosters to offer both public Azure services and their own congruent Azure Stack services in a consistent way while still keeping control of the customer relationship.

For enterprises, using Azure Stack to offer IaaS also makes sense. If you don't offer IaaS in your own datacenter, expect your developers to use public clouds to get it. Even though doing this might violate your organization's security and regulatory policies, the agility this self-service deployment offers is too attractive for them to ignore. Providing it yourself is an obvious way to keep those internal customers using in-house services. And since doing IaaS better than your competitors provides no real advantage, using an off-the-shelf solution, especially one that provides powerful technologies such as ARM, makes sense.

Azure Stack also brings other benefits to enterprises. Its detailed billing lets you do chargeback, if you wish, or at least show your IT consumers exactly what their usage is. Letting your developers create their own VMs with any configuration they want can also improve their productivity significantly. (Doing this at Microsoft had a substantial positive impact on our own development organization.) Azure Stack also provides an insurance policy for Azure development. If it's ever necessary to bring an Azure-native application in house, Azure Stack can make this possible.

Creating PaaS Applications

IaaS is useful, and virtually every cloud platform today provides it. Going forward, however, Microsoft believes that new applications will mostly be written on a PaaS foundation. According to NIST, PaaS means that the "consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications". As described next, Azure Stack supports multiple PaaS options.

An Example

Recall the simple web application shown earlier in Figure 6. Because it was built on IaaS, the creators of this application needed to explicitly create, configure, and manage the VMs it ran in, define the virtual network and load balancer, and specify other low-level aspects of its environment. Building that same application on PaaS is much simpler, as Figure 7 illustrates.

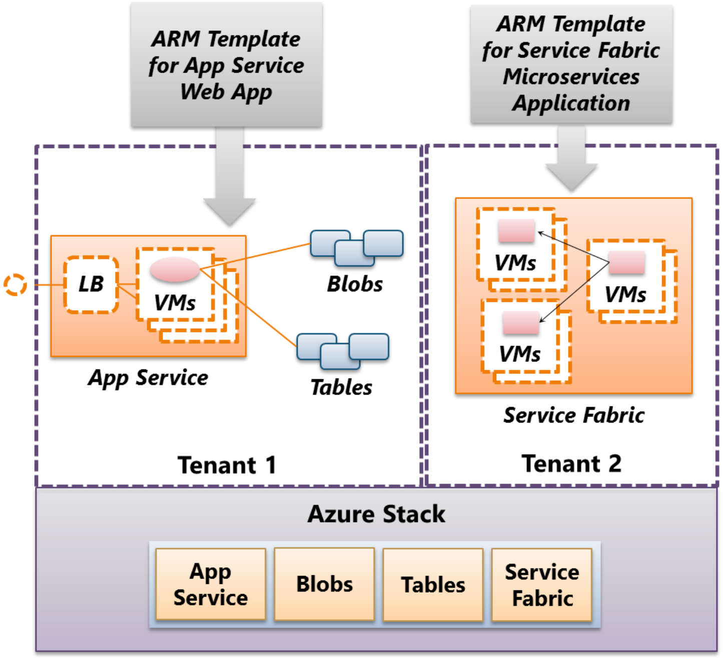

Figure 7: Azure Stack includes various PaaS technologies, making it simpler to create and deploy applications.

This example once again shows two different tenants, each using a different PaaS technology provided by Azure Stack. Tenant 1 implements a traditional web application using App Service. This web app is replicated across multiple VMs and relies on a load balancer. But unlike the IaaS example shown in Figure 6, the creators of the application don't need to worry about these things. They simply provide the application's code to App Service, and this PaaS offering automatically deploys it across the required VMs, handles load balancing, and more. It's much simpler than building the same application using IaaS.

Service Fabric, another Azure Stack PaaS technology, offers similar benefits. A primary difference from App Service, however, is that Service Fabric is designed to support applications built using a microservices architecture. Still, applications built on this foundation don't need to worry about the infrastructure they use—Service Fabric handles this for them. Developers can focus entirely on the application itself, the thing that provides real business value.

Notice also that both of these applications are deployed using ARM templates. Even though PaaS deployment is simpler than IaaS deployment, using ARM still makes sense. Why not automate everything you can?

Why Do This?

Whether you're a service provider or an enterprise, offering PaaS to your cloud consumers makes sense. After all, supporting applications is the real goal of any cloud platform, and PaaS provides a powerful way to do this. Most of the time, there's no real value in letting development teams work directly with VMs, virtual networks, and other IaaS technologies. Letting your Azure Stack administrators handle this, while providing PaaS services to developers optimizes the time and the skill set of both groups.

Because PaaS lets developers create applications more quickly, it's useful for both service providers and enterprises. For service providers, supporting Azure Stack PaaS lets you offer higher-value business solutions. Rather than providing just the traditional IaaS offered by many hosters today, your customers will be able to solve their business problems faster and for less money.

For enterprises, the ability of PaaS to let you build and modify applications more quickly can help you compete with startups who don't have the baggage of traditional IT. It can also improve the way you build and run your applications. Here are some examples:

- App Service provides an IIS-compatible platform for web applications that allows higher density than you probably have today. Moving existing web applications to App Service can let you run these applications more efficiently.
- Suppose you want to build a mobile application with a backend that runs on-premises. App Service includes an option that makes this simple to do. And because this technology is compatible with the global Azure cloud, you can easily move your backend code to the public cloud if it becomes necessary.
- An Internet of Things (IoT) scenario, such as an industrial setting, might require very low latency communication between a device and the code that controls this device. To take advantage of the speedy development and low-cost operation that PaaS provides, you could create the control code on Azure Stack, then run it locally. This gives you the benefit of modern cloud technologies without the delays inherent in communicating across the Internet.
- As business needs change, your organization periodically needs to modernize or perhaps completely re-build existing custom applications. Doing this on a PaaS foundation can help you create these applications faster and run them with fewer management headaches. As more and more organizations confront the challenge of application modernization, PaaS will become the default approach for both private and public clouds.

The PaaS approach also makes using diverse storage options easier. In the App Service example shown in Figure 7, the application relies on Blobs and Tables, two storage technologies that first appeared in Azure. Because they're now in Azure Stack as well, applications created for Azure that use these technologies can be moved unchanged to clouds run by enterprises and service providers. And because both Blobs and Tables are provided as managed services, applications can more easily use the storage approach that best fits their data. (Although it's not shown in Figure 7, App Service on Azure Stack will also support managed relational storage, including SQL Server and MySQL.)

Finally, recognize that Azure Stack's built-in PaaS technologies aren't the only options. You're free to install others or to take advantage of related technologies, such as Docker. To make this easier, Azure Stack provides a marketplace that's similar to the Azure Marketplace. This allows easily selecting and installing other cloud technologies and applications into your Azure Stack cloud.

The Azure Stack Marketplace

The Azure Marketplace lets Microsoft's customers acquire Azure-based solutions. This includes Microsoft technologies, such as preconfigured VMs running SQL Server, along with many open source offerings, such as Linux distributions from Red Hat, Canonical, and others. Other vendors also make their solutions available to deploy on Azure through this marketplace, including DataStax Cassandra, IBM MQ, Barracuda Email Security Gateway, and lots more.

The Azure Stack Marketplace provides something similar for enterprises and service providers. You're free to add your own content, such as locally developed services and applications, to your Marketplace. You can also offer content that's available in the Azure Marketplace, an option referred to as *syndication*. Doing this creates a link between an Azure Marketplace offering and the same offering made available locally by an enterprise or service provider. Among other things, this link ensures that patches made to the Azure Marketplace offering are passed down to the local Azure Stack Marketplace.

In the public cloud, the Azure Marketplace is a central place for users to find and deploy Azure-based services. By providing both Azure content and local content, the Azure Stack Marketplace plays the same role for enterprises and service providers.

Running SaaS Applications

IaaS and PaaS both allow creating custom applications. But most organizations also buy a significant number of packaged software solutions. In the last few years, purchasing these as SaaS rather than on-premises packaged software has become increasingly popular. Yet even though SaaS applications today run nearly exclusively in the public cloud, Azure Stack still has an important role to play.

An Example

One of the challenges for a software vendor creating a SaaS application (especially smaller vendors) is that some customers are all but certain to demand an on-premises version of this application. Just as private clouds are often required to meet security and regulatory requirements, on-premises applications are also often necessary.

But building a scalable, reliable SaaS application can be quite different from building a traditional on-premises application. How can the software vendor hit both targets effectively? One answer is by building the application to run on both Azure and Azure Stack. Figure 8 illustrates how this might look.

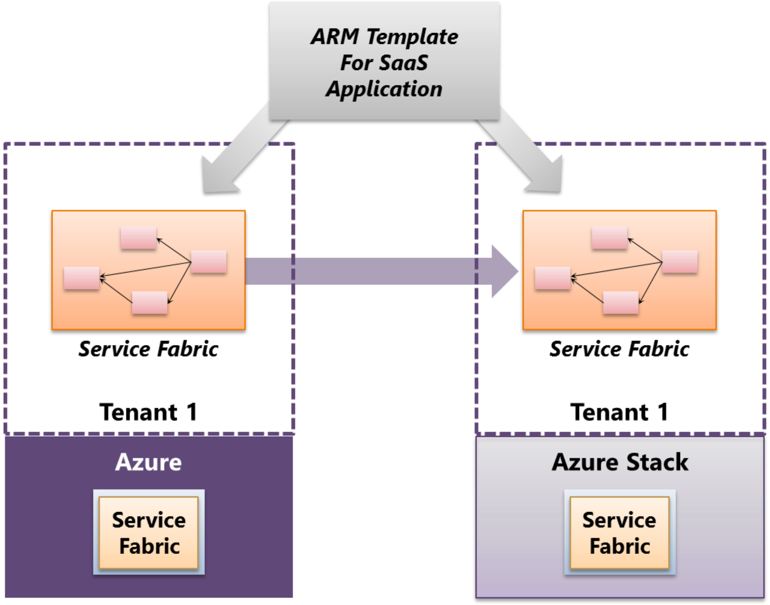

Figure 8: Azure Stack makes it easier to move an Azure SaaS application to an enterprise or service provider running Azure Stack.

SaaS applications need to very scalable, very reliable, and easy to update. To allow these things, it's common to build these applications using a microservices architecture. Service Fabric is designed to support this approach. In fact, Service Fabric underlies a number of Microsoft's own public cloud offerings today.

As the figure shows, a software vendor can create a microservices-based SaaS application using Service Fabric on Azure, along with an ARM template for deploying this application. For customers that need their own in-house version of this application, the software vendor can simply move the software—and the ARM template—to Service Fabric running on Azure Stack.

Why Do This?

The ability to deploy the same SaaS application in a global public cloud and in customer (or service provider) datacenters lets a software vendor kill two birds with one stone. Rather than building a separate version of the application for each of these two environments, the vendor can maintain a single codebase that works in both. This expands their addressable market without the extra overhead of different versions. And because Service Fabric provides a modern microservices platform, the software vendor doesn't have to give up the benefits of this approach.

Even though this example uses Service Fabric, it's not the only option. A software vendor is free to create their SaaS application using other technologies, including App Service, Docker, and more. All of these can run on both Azure and Azure Stack. Whatever option the vendor chooses, ARM templates make it easy to deploy the application in both environments.

Conclusion

Maybe you’re an IT leader who’s already using Azure and wants to create a compatible private cloud environment. Or maybe you’re a service provider who wants to offer Azure services to your customers. For these situations and others, Azure Stack is the answer.

This new offering lets on-premises environments and service providers participate in the innovation and agility that Azure cloud technologies bring. It also has other benefits, including these:

- Application developers can be more productive.
- Applications and data can be moved between Azure Stack and Azure unchanged, for both Windows Server and Linux.
- Managing your infrastructure gets simpler.
- Knowledge of Azure technology is relevant across both private and public cloud.
- Because Azure Stack provides a packaged solution, enterprises can create a private cloud without the complexity of older approaches.
- Internal IT organizations can provide the cloud services their developers demand.
- Service providers can offer leading edge cloud services anywhere in the world.

Some of our competitors want you to go all in with the public cloud. Others think you should avoid the public cloud entirely. At Microsoft, we think it should be up to you to decide what runs in the public cloud and what runs in your own cloud. We also think that you should have the same cloud technologies in both, and that you shouldn’t be locked into either deployment option. Microsoft is the only global cloud vendor that takes this approach, because we believe that applications and data should move easily from one to the other as your needs change.

Cloud is a model, not just a place. This is why we provide both Azure and Azure Stack. Whatever mix of these cloud technologies is right for you, we are committed to helping you succeed.

For More Information

Azure Stack blog: https://azure.microsoft.com/en-us/blog/tag/azure-stack/

ARM templates: https://github.com/Azure/azure-quickstart-templates

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

Some information relates to pre-released product which may be substantially modified before it's commercially released. Microsoft makes no warranties, express or implied, with respect to the information provided here.