



application development specialists

Developing and Deploying Web Applications in Windows Azure

Web Sites vs Web Roles vs Virtual Machines

25/01/2014

Rob Moore | rob.moore@readify.net | 0400 777 763

Readify Pty Ltd ABN 13 097 323 781 | P 1300 666 274 | E info@readify.net | W readify.net

2012 MICROSOFT
PARTNER OF THE YEAR
WINNER



Contents

Executive Summary.....	3
What are Windows Azure Web Sites?	4
What are Windows Azure Web Roles?.....	5
What are Windows Azure Virtual Machines?	6
Advantages and Disadvantages of Azure Web Sites	7
Advantages compared to Web Roles and Virtual Machines	7
Disadvantages compared to Web Roles and Virtual Machines	7
Advantages and Disadvantages of Azure Web Roles	8
Advantages compared to Web Sites and Virtual Machines	8
Disadvantages compared to Web Sites and Virtual Machines.....	8
Advantages and Disadvantages of Virtual Machines.....	9
Advantages compared to Web Sites and Web Roles.....	9
Disadvantages compared to Web Sites and Web Roles.....	9
Deployment	10
Web Sites	10
Web Roles.....	10
Virtual Machines.....	11
Conclusion.....	12
Further Reading	13

Executive Summary

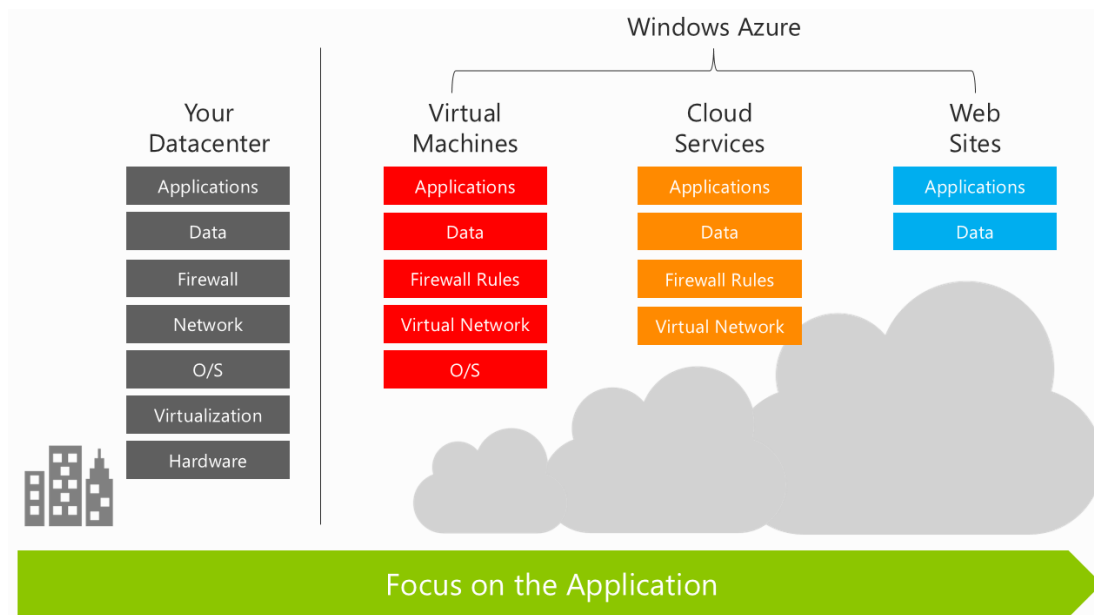
When deploying web applications in Windows Azure there are a number of options available to you. It's not always immediately clear from the documentation which of the options are the right ones to use in any given situation.

This whitepaper will explain the three main options that are available in Windows Azure to deploy web applications and will cover the similarities and differences between them in order to make the decision of which to use easier. It will also cover the different deployment strategies available to you after you have made the decision.

The three main options available to deploy web applications in Azure are:

- Windows Azure Web Sites
- Windows Azure Web Roles (part of Cloud Services)
- Windows Azure Virtual Machines (which can be used to create a web farm)

The following diagram illustrates the high-level difference between the three options¹:



When moving from an on-premise application to Windows Azure's Infrastructure as a Service (IaaS) offering you no longer need to worry about server hardware or the virtualisation layer. When moving from IaaS to Web Roles you no longer need to worry about the Operating System (including OS and IIS patching/updates). When moving from Web Roles to Web Sites you no longer have to worry about networking or firewall constraints (and other similar advanced features) – you just deploy your application and data sources and let Azure initialise and run the application for you. As you move further to the right the management/operational support overhead of the application decreases, but you trade that for a loss in flexibility.

¹ <https://github.com/WindowsAzure-TrainingKit/PRESENTATION-WindowsAzureWebSites>

What are Windows Azure Web Sites?

Windows Azure Web Sites is the simplest and easiest way to deploy web applications in Azure; you can quickly provision a free site, deploy to it almost instantly using a number of different languages and deployment options, quickly and seamlessly configure diagnostics and monitoring and easily scale your site up and out as your application usage and computation complexity grows. All of this functionality is completely driven by user friendly interfaces in the Azure Management Portal that you can log into to administer your Azure applications – you don't need to edit XML configuration files or deploy code to leverage any of this functionality.



Windows Azure Web Sites

powerful web sites in seconds

start simple

start free, scale up and out as you go

code smart

Multiple platforms and SDKs, develop on Windows, OSX or Linux

go live

deploy live in seconds, easily monitor performance, rapidly diagnose and fix issues

2

Web Sites allows you to deploy classic ASP, ASP.NET, python, PHP and nodeJS applications out of the box (as well as any language that can be driven from a FastCGI handler). There is also a gallery of open source applications that you can install into a Windows Azure Web Site in a single click ranging from CMS apps like DotNetNuke, Drupal, Umbraco and WordPress to forum apps like phpBB to wiki apps like MediaWiki.

Web Sites is actually implemented on top of the Cloud Services platform and is a good example of how Microsoft is dogfooding their Azure platform internally³⁴. Web Sites has three tiers available to use:

- Free, which is multi-tenanted and has daily quotas
- Shared, which increases the quotas
- Standard, which allows for the deployment of dedicated VMs with no quotas

To see pricing and quota details of the different tiers consult the Azure website at:

<http://www.windowsazure.com/en-us/pricing/details/web-sites/>.

² <https://github.com/WindowsAzure-TrainingKit/PRESENTATION-WindowsAzureWebSites>

³ <http://msdn.microsoft.com/en-us/magazine/jj883953.aspx>

⁴ <http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B329/>

What are Windows Azure Web Roles?

Windows Azure Cloud Services is a Platform as a Service offering that consists of two types of machines:

- Web Roles: A virtual machine with IIS configured and the ability to run tasks in the background
- Worker Roles: A virtual machine without IIS configured that can run tasks in the background

Both roles allow you to configure (via API calls, XML configuration and / or portal UI interaction):

- IIS sites
- Load balancer
- Firewalled ports
- VM sizes
- Number of instances
- Certificates
- Diagnostics
- Remote Desktop
- File system mounts (into Blob storage)
- Caching nodes (In-Role cache)
- Virtual Network endpoints

Cloud Services are packaged along with their configuration by a Visual Studio integrated SDK and you can combine multiple roles into a single deployment. Each deployment will have a separate Visual Studio project associated with it that defines the make-up of the deployment (e.g. web role implemented by web project X, worker role implemented by class library Y as well as configuration files for both web and worker role).

Cloud Services was one of the first components released in Windows Azure and is quite a mature platform. The information that is available about its architecture is slightly out of date⁵, but is mostly correct since there hasn't been many changes to Cloud Services in the last few years.

There are a number of sizes of machine available for Cloud Services that allow you to tweak the performance you get (and by association the cost) based on your application requirements; this ranges from Extra Small instances with a shared CPU core and 768MB of RAM through to heavy-workload machines like the A7 size with 8 dedicated CPU cores and 56GB of RAM. For latest pricing details and size specifications consult the Azure website at <http://www.windowsazure.com/en-us/pricing/details/cloud-services/>.

⁵ <http://blogs.msdn.com/b/kwill/archive/2011/05/05/windows-azure-role-architecture.aspx>

What are Windows Azure Virtual Machines?

Windows Azure Virtual Machines allows you to provision VMs with persistent virtual hard drives (VHDs) and you have full control over everything on that VM from the operating system and associated patching and updating through to the applications and software you deploy on the VM. Both Windows and Linux VMs are supported and while you can use any VHD you want⁶⁷ there are also a number of pre-created images you can use to get up and running quickly, including (as at January 2014):

- Windows Server 2008/2012
- SharePoint 2013
- SQL Server 2014/2012/2008
- BizTalk Server 2013
- openSUSE 13
- SUSE Linux Enterprise 11
- Ubuntu 12/13/14
- CentOS 6 (OpenLogic)
- Oracle Linux 6.4
- Oracle Database 12
- Oracle Web Logic 12
- Visual Studio 2013
- JDK 6/7

You can use PowerShell Remoting / SSH to completely programmatically manage your Windows/Linux VMs remotely and you can use the Windows Azure API to programmatically provision the VMs in the first place. This gives you ultimate flexibility and means you can do practically anything you want using them.

If you are provisioning Windows VMs then it's definitely worth looking at Boxstarter⁸, which has first class support for Windows Azure Virtual Machines⁹.

The VM size options are similar to Cloud Services with the pricing being similar (slightly expensive, as at August 2013) too. For the latest pricing details and size specifications consult the Azure website at <http://www.windowsazure.com/en-us/pricing/details/virtual-machines/>.

Using Windows Azure Virtual Machines you can set up a web farm to host your web applications in the same way that you would for an on-premise solution.

⁶ <http://www.windowsazure.com/en-us/manage/windows/common-tasks/upload-a-vhd/>

⁷ <http://www.windowsazure.com/en-us/manage/linux/common-tasks/upload-a-vhd/>

⁸ <http://boxstarter.org/>

⁹ <http://www.mattwrock.com/post/2014/01/21/Released-Boxstarter-23-Windows-Azure-integration.aspx>

Advantages and Disadvantages of Azure Web Sites

Advantages compared to Web Roles and Virtual Machines

There are a number of out-of-the-box features that are possible in Web Roles and Virtual Machines, but that are much more easily/quickly accomplished in Web Sites:

- Near-instant Git, TFS, FTP, Web Deploy and Dropbox deployments
- One-click roll-backs
- Multiple websites on a single farm (unlike Web Roles by default)
- Simpler VS solution (no extra project like in Web Roles)
- Multiple programming languages supported (e.g. Python, nodeJS etc.)
- Scheduling continuous or periodic background jobs in a number of languages (Web jobs)
- Always On Support (keeping your application pool constantly warm)

There are also a number of core advantages that are unique to Web Sites:

- Fast (read: seconds) provisioning and scaling of new nodes in the web farm
- One-click blogs and CMSs
- No effort monitoring and diagnostics
- Deploy time config changes
- A-record DNS support (aka naked domains without "www")
- Memory dumps via REST API
- One-click install of New Relic monitoring
- Free SSL (if using azurewebsites.net subdomain)
- Shared file system across all instances allowing you to deploy apps like Umbraco
- Staging environment with custom DNS name (Web Roles has <guid>.cloudapp.net)
- Automatic by-the-minute diagnostics data on the Windows Azure portal dashboard
- You are eligible for the SLA with only one instance

Disadvantages compared to Web Roles and Virtual Machines

- You can't use ports other than 80/443
- You can't remote desktop into your VM
- You can't get network isolation
- You can't execute arbitrary start up scripts
- You can't use elevated privileges
- You can't mount from blob storage
- You have limited IIS configuration
- You have slightly limited autoscaling capabilities
- You don't have the XS, XL, A6 and A7 VM sizes
- Custom domain SSL costs extra
- No ability to upload custom certificates
- You still get charged when your website is stopped
- Not currently available in all sub-regions
- You can't use traffic manager to improve performance, availability and resiliency of your application across multiple geographic regions
- The SLA is slightly lower (99.9%) than Web Roles and Virtual Machines with 2 instances (99.95%)

Advantages and Disadvantages of Azure Web Roles

Advantages compared to Web Sites and Virtual Machines

- Flexible diagnostics capabilities
- Complex autoscaling capabilities
- Huge scale (100s of VMs or more)
- Complex configuration via XML
- Affinity Groups to improve performance
- Upgrade/Fault domains
- You can host files in the CDN from the website itself

Disadvantages compared to Web Sites and Virtual Machines

- Even though the speed has improved over time, out of the box deployments for Cloud Services are still very slow compared to Web Sites (8+ minutes)
- The Visual Studio solution is more complex because you need to have an extra "Cloud" project
- The Cloud Configuration files that come with Cloud Services can cause confusion since it provides two ways to configure your application: web.config and the cloud configuration. Cloud configuration is relatively easy to change at runtime, which avoids a long deployment time to change the config, but potentially results in losing track of configuration information that would otherwise be modified in source control before deploying

Advantages and Disadvantages of Virtual Machines

Advantages compared to Web Sites and Web Roles

- Complete flexibility/control
- Easy to deploy legacy applications without needing to re-architect
- Ability to configure IP address ACLs at a load balancer endpoint level (however you can configure at an IIS level in Web Sites / Web Roles and Windows level in Web Roles)

Disadvantages compared to Web Sites and Web Roles

- Scaling the web farm is difficult and requires a lot of custom script development
- There is no out of the box auto-scaling solution
- You need to have operational staff that can manage the Operating System and Web Server and ensure appropriate patching/updating etc.
- No in-built support for debugging your applications from Visual Studio (although you can use the usual techniques for debugging remote machines)
- No built-in support for performing a quick staging <-> production type deployment (with ability to quickly rollback; see VIP swap for Web Roles and staging support for Web Sites)

Deployment

Web Sites

One of the biggest advantages with Windows Azure Web Sites is the ease with which you can deploy your applications out of the box. There are a range of different deployment options available to you, including:

- FTP
- Web Deploy
- TFS
 - Team Foundation Service
 - CodePlex
- Mercurial
 - Publicly accessible Hg repository
 - Bitbucket (public or private)
 - CodePlex (public or private)
- Git
 - Hosted Git repository
 - Publicly accessible Git repository
 - GitHub (public or private)
 - Bitbucket (public or private)
 - CodePlex

Web Roles

When deploying Web Roles you have three main options after creating your deployment package (using Visual Studio or MSBuild) – you can manually upload the package via the Windows Azure Portal, upload the package using Visual Studio or you can programmatically deploy the package using the API (by making the HTTP calls directly¹⁰, using the PowerShell commandlets¹¹ or using the Management API dlls from NuGet¹²).

When deploying a Cloud Services deployment you will need to wait anywhere from 8 to 12 minutes depending on the complexity of your application. If you want to practice regular deployments then this is a prohibitively large amount of time to wait (particularly when compared to the near instant deployment you get out of the box with Web Sites).

Rather than tying Web Role provisioning to your application deployment you can instead treat the Web Roles as infrastructure and provision a load-balanced web farm that you are able to deploy your applications to in a similar way you treat the Azure Web Sites service. This can be tricky with the way Web Roles function (when they recycle or restart all state on the VM is lost), but there are some libraries that exist to allow you to create web farms using Web Roles:

The AzureWebFarm library which creates a persistent, web-deploy enabled web farm and includes a background worker execution model: <https://github.com/MRCollective/AzureWebFarm>

¹⁰ <http://msdn.microsoft.com/en-us/library/windowsazure/ee460799.aspx>

¹¹ <http://msdn.microsoft.com/en-us/library/windowsazure/jj156055.aspx>

¹² <http://www.bradygaster.com/post/getting-started-with-the-windows-azure-management-libraries>

The AzureWebFarm.OctopusDeploy library which creates a web farm that can be connected to an OctopusDeploy server to manage deployments of web sites or windows services:

<https://github.com/MRCollective/AzureWebFarm.OctopusDeploy>

Virtual Machines

When using a web farm constructed using Virtual Machines your deployment options are really no different to an on-premise web farm. The following are the most obvious deployment strategies for .NET applications:

- Manual deployment – not really a feasible option; **not recommended**, but always an option
- Shared drive – create a network share across your farm and deploy to the master node – **not recommended** because it's not as scalable
- Web Deploy to each server individually – **not recommended** because of the change of an inconsistent deployment and it's not scalable
- IIS Web Farm Framework – it hasn't yet been updated to work with IIS 8 and WebDeploy 3 and there is no documentation about setting it up in Azure, but it should be no different than setting it up on-premise; will require an extra VM to act as a control server
- Use Windows Azure Pack to get an Azure Web Sites like experience
- Web Deploy to one server using MsDeploy and use DSFR syncing across the farm to keep all nodes up to date – requires all servers in the farm to be added to an Active Directory forest
- OctopusDeploy – will require an extra VM to act as a central server; **this is the most recommended solution**

Conclusion

If you are running existing enterprise server applications in Azure such as SQL Server, SharePoint Server or Active Directory then you don't have any choice other than using Virtual Machines. Similarly, when deploying legacy web applications that aren't compatible with Web Sites or Web Roles and you aren't in a position to re-architect them then Virtual Machines are your only option. The same thing applies if you have a Linux web application or want to have full control over the operating system (and have the operational staff available that can maintain the servers).

Otherwise, if you have an application that needs In-role cache nodes, CDN on files stored in the website (rather than just blob storage), large amounts of memory, extreme scale, network isolation, geo-redundancy or scaling, deployment to SE Asia for latency reasons or requires RDP, elevated permissions advanced IIS configuration, complex diagnostics or cost-effectiveness for sometimes-on scenarios then Web Roles are your best choice.

In most other scenarios Web Sites is the best choice since it's the simplest option and is fully-managed. The Web Sites platform is being constantly innovated on with new features surfacing almost every month – Microsoft are investing a lot of resources into making sure it's a state of the art platform. If your application grows in complexity over time and needs to migrate from Web Sites to Web Roles or Web Roles to Virtual Machines this is a relatively easy process so you should be able to start with the simplest option initially (Web Sites) with confidence you can change later if required.

Another way to think of the three options are:

- Virtual Machines are the Infrastructure-as-a-Service (IaaS) offering in Azure
- Web Roles are the Platform-as-a-Service (PaaS) offering in Azure
- Web Sites are akin to the traditional shared hosting option that have been available for years (except on steroids)

Further Reading

- <http://robdmoore.id.au/blog/2012/06/09/windows-azure-web-sites-vs-web-roles/>
- <http://www.windowsazure.com/en-us/manage/services/web-sites/choose-web-app-service/>
- <http://michaelwasham.com/2012/06/08/understanding-windows-azure-virtual-machines/>
- <http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B329/>
- <http://www.windowsazure.com/en-us/documentation/services/cloud-services/>
- <http://www.windowsazure.com/en-us/documentation/services/web-sites/>
- <http://www.windowsazure.com/en-us/documentation/services/virtual-machines/>
- <http://www.troyhunt.com/2014/01/with-great-azure-vm-comes-great.html>