Microsoft

Introducing Windows Azure

For IT Professionals

Mitch Tulloch with
the Windows Azure Team

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2013949894
ISBN: 978-0-7356-8288-7

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at *http://www.microsoft.com/learning/booksurvey*.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Valerie Woolley
Editorial Production: Christian Holdener, S4Carlisle Publishing Services
Copyeditor: Andrew Jones

Contents

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Introduction

The Windows Azure public cloud platform is one of the three pillars of Microsoft's Cloud OS vision that will transform the traditional datacenter environment, help businesses unlock insights in data stored anywhere, enable the development of a wide range of modern business applications, and empower IT to support users who work anywhere on any device while being able to manage these devices in a secure and consistent way. The other two pillars of the Cloud OS are, of course, Windows Server 2012 R2 and Microsoft System Center 2012 R2, and Microsoft Press will soon be releasing free Introducing books on these platforms as well.

What's really exciting to me personally as an IT pro is that this is all coming at once. General Availability (GA) of these latest versions of Windows Server and System Center is currently scheduled for October 18, 2013, which is less than two months away as this book is being written. In sync with these two releases, the Windows Azure platform has also been enhanced in recent months with preview releases of new services like Windows Azure BizTalk Services, Windows Azure Traffic Manager, and Windows Azure HDInsight. And in the same timeframe, services that were previously in preview like Windows Azure Web Sites and Windows Azure Mobile Services have now reached the GA milestone.

In fact, as I write this Introduction (it happens to be the last piece of the book that I'm writing) I just noticed that another new service, Windows Azure Store, has just entered preview. Fortunately, it turns out that my free Windows Azure subscription as an MSDN subscriber currently doesn't support purchasing from the Store in my geographical region, so I can't test this preview feature just yet. I'm actually glad about this because I just finished writing the last chapter and don't want to go back and have to revise it again!

My point of course is that Windows Azure, the public cloud portion of Microsoft's Cloud OS, is a constantly evolving platform with new features entering preview all the time. One has to draw a line somewhere though, so we've decided to title this book *Introducing Windows Azure for IT Professionals* as it tries to capture the essence of what Windows Azure can do for your business as Microsoft's Cloud OS vision becomes a reality with the release of Windows Server 2012 R2 and System Center 2012 R2.

Whether you're new to the Windows Azure platform or are already using it in your business, this book has something that should interest you. Most Windows Azure services are described in some detail, with screenshots used to demonstrate some of the multitude of capabilities of the platform. And for the experienced we have lots of under-the-hood insights and expert tips written by Microsoft insiders who develop, test, and use the Windows Azure platform.

So whatever your goals are in reading this book, you're going to find new things about the Windows Azure platform that will amaze and delight you. Because, as you'll soon see in Chapter 1, Windows Azure can be anything you want it to be!

About the companion content

The companion content for this book can be downloaded from the following page:

http://aka.ms/IntroAzure/files

The companion content includes the Windows PowerShell script and some code samples from sidebars in chapters 2 and 4 of this book.

Acknowledgments

Three groups of people have helped make this book possible, and as author I'd like to thank them all here.

First, the following experts at Microsoft have contributed sidebars that explain and demonstrate different aspects of the Windows Azure platform:

- Ashish Goyal
- Brian Hitney
- Clemens Vasters
- Eric Mattingly
- Hai Ning
- Hanu Kommalapati
- John Savill
- Mike Gaal
- Nikhil Sachdeva
- Paulo Almeida Tanaka
- Robert Nottoli
- Yuri Diogenes

Second, the following Microsoft insiders have peer reviewed the content of this book to help us ensure it's as accurate as possible:

- Hanu Kommalapati
- Ivan Aspillaga Velasco
- Manoj K Jain
- Simon Hart
- Steve Espinosa
- Tal Ben-Shalom

Finally, I'd also like to thank Valerie Woolley, Content Project Manager at Microsoft Press; Christian Holdener at S4Carlisle Publishing Services; and copyeditor Andrew Jones.

Errata & book support

We've made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site at oreilly.com:

http://aka.ms/IntroAzure/errata

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://www.microsoft.com/learning/booksurvey

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

Understanding Windows Azure

This chapter introduces the Windows Azure platform and describes the different services and types of solutions that this platform makes possible. We've included this chapter mainly for readers who are not yet familiar with what Windows Azure is and what it can do. If you are already familiar with Windows Azure, feel free to skip this first chapter, but you might want to read it anyways as you might learn something valuable that can help your business grow and meet the challenges it faces in the marketplace.

What is Windows Azure?

If you ask, "What is Windows Azure?" the best answer might simply be this:

Windows Azure can be anything you want it to be.

As a cloud platform from Microsoft that provides a wide range of different services, Windows Azure lets you build, deploy, and manage solutions for almost any purpose you can imagine. In other words, Windows Azure is a world of unlimited possibilities. Whether you're a large enterprise spanning several continents that needs to run server workloads, or a small business that wants a website that has a global presence, Windows Azure can provide a platform for building applications that can leverage the cloud to meet the needs of your business.

That's a pretty big assertion to make. How can we know it's true?

Let's examine the facts!

Why use the cloud?

Businesses generally consider moving their applications to the cloud for one of three reasons: speed, scale, and economics. Let's briefly examine each of these advantages that cloud computing can provide.

Speed

Developing applications that run in the cloud can be significantly faster than traditional application development for two reasons. First, you don't have to deploy, configure, and maintain the underlying computing, storage, and networking infrastructures on which

your applications will run. Instead, you can utilize infrastructure resources provided to you by your cloud hosting provider. A good analogy is lighting up your house. If you need to not only purchase and install light bulbs but also pull wires, install switches, buy and set up a generator, and purchase gasoline for your generator, it's going to take a while until you get your house lit up. But if your light bulbs can leverage the wiring installed by the contractor who built your house and electricity provided by your city's power station, you'll be able to get your house lit up in next to no time.

A second reason why cloud-based applications can be faster to deploy has to do with how applications are developed. In a typical enterprise environment, developers create and test their applications in a test environment that only partially simulates the final production environment. For example, an application might be developed and tested on a nonclustered host for eventual deployment onto clustered hosts. Mismatches like this between your development and production environments can slow the development process for business applications because certain problems might be missed in testing and only become apparent when the applications are deployed to production, which might necessitate further testing and development until the applications are behaving as intended. With the cloud, however, you can perform your development and testing in the same kind of environment that your applications will be deployed upon—the computing, storage, and networking resources provided to you by your hoster. This can make testing applications simpler and more reliable, thereby reducing the time to deployment.

Scale

Cloud applications can also scale out quickly because commodity compute, storage, and networking resources are pooled by the hoster and can be provisioned to the tenant (your business) as the need arises. Does your application need more compute resources to meet increasing demand from customers? Running it in the cloud can help keep your customers happy. Does a downturn in the market mean that you don't need all that compute capacity your hoster is providing for your applications? Just scale down how much of your hoster's compute capacity you are using.

In addition to scaling capacity up or down quickly, cloud computing can also provide your applications with global scale and reach for extending your business presence around the world. Such scalability can be essential for businesses to remain competitive in today's global economy.

Economics

Regarding economic efficiencies, many organizations decide to make the move to the cloud because the cost of running their business applications in the cloud can be significantly lower than running them on-premises. By utilizing a pay-for-what-you-use model for cloud computing, your business only pays for the resources it actually consumes. The ability to rapidly and easily scale capacity up or down that the cloud offers makes this approach possible and can help organizations save money. In fact, some Windows Azure services,

such as Windows Azure Virtual Machines, now support per-minute billing to help customers minimize the cost of running their server workloads in Windows Azure.

Cloud computing can also makes economic sense for businesses when it comes to hardware acquisition costs. Traditionally, large enterprises have often purchased and deployed storage area networks (SANs) from third-party SAN vendors to meet their escalating data storage needs. By provisioning storage resources from a cloud hosting provider instead, these enterprises can often significantly reduce their storage acquisition and maintenance costs.

Resisting the cloud

If running business applications in the cloud has so many advantages, why doesn't every business do it? There are several possible reasons for this:

- Companies that have made large investments in traditional IT infrastructure might prefer to have those investments earn out than adopt the cloud computing approach.
- Enterprises that have a complex, heterogeneous IT landscape might find it challenging to integrate cloud computing into their existing infrastructure.
- The reliability of the hosting provider is another issue for many companies. Lack of knowledge about the infrastructure used by the hoster for providing cloud services to customers can make some organizations reluctant to risk a move to the cloud.
- Business politics can also cause a company to resist migrating their applications to the cloud. The traditional in-house IT culture of control can make it difficult to embrace the cloud.
- Organizations that have special requirements in the areas of security and compliance might have legal or regulatory requirements that block them from implementing different forms of cloud computing.

As you will see later in this chapter, however, Windows Azure can address these different concerns about existing IT investments, integration challenges, reliability, control, security, and compliance so that your business can reap the benefits of speed, scale, and cost savings that cloud computing can provide. But first let's take a closer look at what Windows Azure is and what it can do for your business.

Are you ready for the cloud?

Not every business is ready to take advantage of the different types of cloud computing services available. To help you learn whether your business is ready for the cloud, Microsoft has a web-based Cloud Security Readiness tool that assesses the systems, processes, and productivity of your current IT environment and generates a custom noncommercial report that provides recommendations to help you evaluate the benefits of cloud computing. To use this free tool, go to *http://technet.microsoft.com/en-us/security/jj554736.aspx*.

Windows Azure under the hood

We said earlier that if we ask the question, "What is Windows Azure?" the best answer might be this:

Windows Azure is anything you want it to be

Of course this kind of definition doesn't really explain what Windows Azure is, so let's look at the definition that Microsoft uses for describing Windows Azure:

> *Windows Azure is an open and flexible cloud platform that enables you to quickly build, deploy, and manage applications across a global network of Microsoft-managed datacenters. You can build applications using any language, tool, or framework. And you can integrate your public cloud applications with your existing IT environment.*

This definition tells us that Windows Azure is a cloud platform, which means you can use it for running your business applications, services, and workloads in the cloud. But it also includes some key words that tell us even more:

- **Open** Windows Azure provides a set of cloud services that allow you to build and deploy cloud-based applications using almost any programming language, framework, or tool.
- **Flexible** Windows Azure provides a wide range of cloud services that can let you do everything from hosting your company's website to running big SQL databases in the cloud. It also includes different features that can help deliver high performance and low latency for cloud-based applications.
- **Microsoft-managed** Windows Azure services are currently hosted in several datacenters spread across the United States, Europe, and Asia. These datacenters are managed by Microsoft and provide expert global support on a 24x7x365 basis.
- **Compatible** Cloud applications running on Windows Azure can easily be integrated with on-premises IT environments that utilize the Microsoft Windows Server platform.

Windows Azure Content Delivery Network

In addition to the Microsoft-managed datacenters where Windows Azure runs, Windows Azure also has a Content Delivery Network (CDN) currently located at 24 different sites in the United States, Europe, Asia, Australia, and South America. This CDN consists of a geographically distributed system of servers that enables Windows Azure to deliver high-bandwidth content for performance-sensitive customers by caching data, images, pictures and video of Windows Azure compute instances. Examples of cachable content include web objects, downloadable objects (such as media files, software, and documents), applications, real-time media streams, database queries, DNS traffic, routes, and more. Microsoft guarantees that the Windows Azure CDN will successfully respond to client requests and deliver the

requested content without error with 99.9 percent reliability. The results of using the CDN to cache Windows Azure data include:

- **Better use experience for users who are located far from the content source.**
- **Improved application performance when multiple "trips back and forth across the Internet" are required to load content for an application.**
- **Greater ability for organizations to handle sudden increases in traffic such as during a product launch event.**

CDN is available to customers as an add-on feature for Windows Azure subscriptions. It has a separate billing plan and can be configured using the Windows Azure Management Portal. For more information on how to set up CDN, see *http://msdn.microsoft.com/en-us/library/windowsazure/ee795176.aspx*.

Windows Azure services

Windows Azure provides businesses with four basic categories of cloud-based services:

- Compute services
- Network services
- Data services
- App services

Let's take a quick look at the different Windows Azure services that are currently available in this Fall 2013 release of the Windows Azure platform. Then, in the chapters that follow this one, we'll dig deeper into each of these services so you can understand more clearly what they are, how they work, and what they're capable of doing for your business.

NOTE **The number of different Windows Azure services available is expanding, with new services continually being added and existing services enhanced as the Windows Azure platform continues to evolve. For some fascinating insight into how these services have evolved over time, see the sidebar titled "The evolution of Windows Azure" later in this chapter.**

Compute services

Windows Azure compute services provide the processing power required for cloud applications to be able to run. Windows Azure currently offers four different compute services:

- **Virtual Machines** This service provides you with a general-purpose computing environment that lets you create, deploy, and manage virtual machines running in the Windows Azure cloud.

- **Web Sites** This service provides you with a managed web environment you can use to create new websites or migrate your existing business website into the cloud.
- **Cloud Services** This service allows you to build and deploy highly available and almost infinitely scalable applications with low administration costs using almost any programming language.
- **Mobile Services** This service provides a turnkey solution for building and deploying apps and storing data for mobile devices.

Network services

Windows Azure network services provide you with different options for how Windows Azure applications can be delivered to users and datacenters. Windows Azure currently offers two different network services:

- **Virtual Network** This service allows you to treat the Windows Azure public cloud as if it is an extension of your on-premises datacenter.
- **Traffic Manager** This service allows you to route application traffic for the user who is using the application to Windows Azure datacenters in three ways: for best performance, in round robin fashion, or using an Active/Passive failover configuration.

Data services

Windows Azure data services provide you with different ways of storing, managing, safeguarding, analyzing, and reporting business data. Windows Azure currently offers five different data services:

- **Data Management** This service lets you store your business data in SQL databases, either with dedicated Microsoft SQL Server virtual machines, using Windows Azure SQL Database, using NoSQL Tables via REST, or using BLOB storage.
- **Business Analytics** This service enables ease of discovery and data enrichment using Microsoft SQL Server Reporting and Analysis Services or Microsoft SharePoint Server running in a virtual machine, Windows Azure SQL Reporting, the Windows Azure Marketplace, or HDInsight, a Hadoop implementation for Big Data.
- **HDInsight** This is Microsoft's Hadoop-based service which brings a 100 percent Apache Hadoop solution to the cloud.
- **Cache** This service provides a distributed caching solution that can help speed up your cloud-based applications and reduce database load.
- **Backup** This service helps you protect your server data offsite by using automated and manual backups to Windows Azure.
- **Recovery Manager** Windows Azure Hyper-V Recovery Manager helps you protect business critical services by coordinating the replication and recovery of System Center 2012 private clouds at a secondary location.

App services

Windows Azure app services provide you with ways of enhancing the performance, security, discoverability, and integration of your cloud apps that are running. Windows Azure currently offers seven different app services:

- **Media Services** This service allows you to build workflows for the creation, management, and distribution of media using the Windows Azure public cloud.
- **Messaging** This consists of two services (Windows Azure Service Bus and Windows Azure Queue) that allow you to keep your apps connected across your private cloud environment and the Windows Azure public cloud.
- **Notification Hubs** This service provides a highly scalable, cross-platform push notification infrastructure for applications running on mobile devices.
- **BizTalk Services** This service provides Business-to-Business (B2B) and Enterprise Application Integration (EAI) capabilities for delivering cloud and hybrid integration solutions.
- **Active Directory** This service provides you with identity management and access control capabilities for your cloud applications.
- **Multifactor Authentication** This service provides an extra layer of authentication, in addition to the user's account credentials, in order to better secure access for both on-premises and cloud applications.

Understanding cloud service models

From a customer perspective, the various kinds of services that cloud computing platforms such as Windows Azure can provide can be broadly lumped together into three categories called service models:

- **Infrastructure as a Service (IaaS)** In this approach, the customer pays the hoster to run a virtual machine in the hoster's cloud. The customer is responsible for configuring and managing the virtual machine's guest operating system and applications. This model views the customer as the IT owner since the customer has complete control over what they can do with their virtual machine.
- **Platform as a Service (PaaS)** In this approach, the customer develops and deploys applications for a specific application architecture. The hoster provides the application runtime, storage, and integration needed to run the customer's application and is responsible for keeping the environment up and running, operating systems updated, and customer data safe. This model views the customer as the application owner since the customer is responsible for developing and maintaining the application. The customer is also responsible for data integrity and business logic.

- **Software as a Service (SaaS)** In this approach, the customer utilizes standardized cloud-based services such as document management or email that are provided by the hoster. This model views the customer as the user who consumes cloud applications, typically as a pay-as-you-go service.

In the Windows Azure platform, Virtual Machines is an example of the IaaS approach while Cloud Services is an example of the PaaS approach. Other Microsoft services like Office 365 are examples of the SaaS approach. Many of the Windows Azure services don't fit within the framework of this simple model, however, and instead provide the "plumbing" that can be used to build and deploy innovative new forms of cloud solutions.

Windows Azure components

The above categorization of Windows Azure services into four types (compute, network, data, and app) isn't the only way that the architecture of Windows Azure can be envisioned. Figure 1-1 shows another way of breaking down the platform into different components.

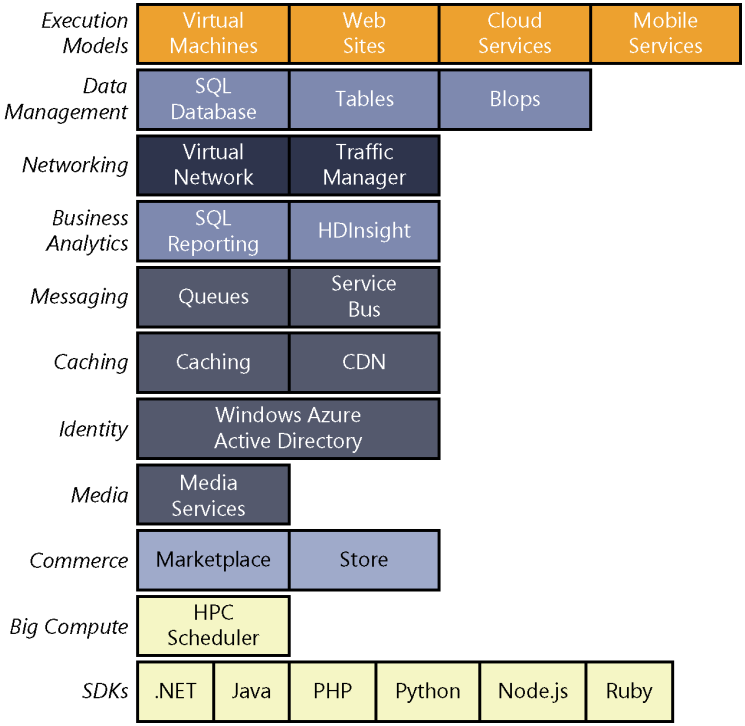


FIGURE 1-1 Another way of viewing the components of the Windows Azure platform.

At the top are the various execution models, namely the different technologies that can be used to execute applications running in the Windows Azure cloud. The execution models correspond to the four Windows Azure compute services of Virtual Machines, Web Sites, Cloud Services, and Mobile Services.

At the bottom are the language-specific software development kits (SDKs) you can use for building, deploying, and managing applications that can run on Windows Azure. The SDKs that are currently supported include ones for .NET, Java, PHP, Node.js, Ruby, and Python. There is also a general Windows Azure SDK that provides basic support for any programming language, for example C++. The latest version (2.1) of the Windows Azure SDK for .NET include SDKs, basic tools, and extended tools for Microsoft Visual Studio 2010 SP1, Visual Studio 2012, and Visual Studio 2013 Preview. See the "Learn more" section at the end of this chapter to find out where you can download this SDK.

MORE INFO **For a pictorial view of the architecture of the Windows Azure platform that you can print and hang on your office wall, download the Windows Azure poster from the Microsoft Download Center at *http://www.microsoft.com/en-us/download/details.aspx?id=35473*.**

The evolution of Windows Azure

The following is a brief year-by-year account of some interesting milestones in the evolution of Windows Azure from a couple of insiders at Microsoft.

2006

Cloud OS Dream Team formed to develop the "RedDog" Operating system spearheaded by Amitabh Srivastava and Dave Cutler.

Azure development starts with three different teams working independently:

- **A secret project called RedDog is started by the Windows team.**
- **The Indigo team (commonly known as Windows Communication Foundation) was working on an incubation feature around communication relay.**
- **The SQL team was working on enabling their services available in the Internet.**

Each of these teams have their own management and infrastructure processes.

The now-called "Windows Azure Service Bus" runs on a single computer under someone's desk.

2007

The teams start to collaborate with each other but still have their own management and infrastructure processes.

Service Bus runs on two computers in a lab.

2008

The three independent teams now work together to form the basis for the Windows Azure platform, these include:

- .NET Services include: Service Bus, Access Control, and Workflow Services
- Live Services: Consistent way for users to store, share, and synchronize documents, photos, files, and information across their PCs, phones, PC applications, and websites
- SQL Services: includes database services and reporting.

Alongside Microsoft SharePoint Services and Dynamic Services, support for an identity framework "Geneva" is also announced.

October: Windows Azure CTP announced in PDC 2008 with the launch of Live Services, Microsoft .NET Services, and Microsoft SQL Services. The users get 2000 VM hours, 50GB of storage, and 20GB of bandwidth.

October: The first SDK released at PDC 2008 with support for Medium Trust Code, BLOBs, Queues, Tables, and Automatic Service Management.

2009

March: SDK CTP 3 with support for Full Trust and Native code releases, support for Geo Location and Fast-CGI announced at MIX 2009.

May: Windows Azure gets recognized as an open platform; PHP, Java SDK CTP announced.

October: Silverlight-based Azure management portal launched for integrated management portal for Cloud Services (PaaS), Service Bus, and SQL Azure.

November: SDK 1.0, first official production SDK is released, it comes with an array of features like Windows Azure Diagnostics and Inter-role communication. CDN and custom domain support for BLOB storage.

November: Microsoft launches BizSpark program to help startups and software entrepreneurs onboard Windows Azure.

2010

February: General availability (GA) of Windows Azure and SQL Azure announced for 21 countries.

June: SQL Azure increases database size support to 50 GB.

July: Microsoft passes 10,000 customers using Windows Azure.

October: Windows Azure Virtual Machine role is announced, which is a nonpersistent Virtual Machine role for deploying custom VHD images.

October: Windows Azure Connect is announced, which enables access to on-premises resource for hybrid applications.

November: Windows Azure SDK 1.3 releases with .NET 4.0, full IIS, and remote desktop support.

2011

Windows Azure Traffic Manager announced to enable control of distribution of user traffic in Windows Azure Cloud Services.

SQL Azure Reporting announced to leverage Microsoft SQL Server Reporting Services on the Windows Azure platform.

HPC Scheduler SDK announced to support compute-intensive, parallel applications.

Windows Azure obtains ISO 27001 certification for its core services.

2012

February: Azure services go down for 23 hours. This day is marked as "Leap Day" in Azure history.

June: Azure Virtual Machine preview announced, which launches Windows Azure as an IaaS vendor.

June: Windows Azure management portal revamped, no Silverlight required. Support for multiple HTML 5 supported browsers. Windows Azure supports Linux VM for allowing users to run OpenSUSE 12.1, CentOS 6.2, Ubuntu 12.04, and SUSE Linux Enterprise Server 11 SP2 using Azure Virtual Machines

June: Continued support for cross-platform languages, Python SDK released, Windows Azure plugin for Eclipse with Java released.

July: Windows Azure Pack (Katal) preview released for Virtual Machine and Web Site hosting in Windows Server.

Windows Azure obtains SSAE 16 / ISAE 3402 attestation.

Windows obtains HIPAA BAA (Business Associate Agreement) compliance (see *http://blogs.msdn.com/b/windowsazure/archive/2012/07/25/security-privacy-amp-compliance-update-microsoft-offers-customers-and-partners-a-hipaa-business-associate-agreement-baa-for-windows-azure.aspx* for more information).

Windows Azure datacenters available in 8 regions:

- North Central US—Chicago, Illinois
- South Central US—San Antonio, Texas
- West US—California
- East US—Virginia

- North Europe—Dublin, Ireland
- West Europe—Amsterdam, Netherlands
- East Asia—Hong Kong
- Southeast Asia—Singapore

2013

April: Windows Azure and related software surpass $1 billion in annual sales.

April: GA for Azure Infrastructure Services with support for large VM sizes and additional preconfigured VM templates.

April: GA for Windows Azure Active Directory (WAAD)

June: GA for Azure Mobile Services in three tiers—Free, Standard, and Premium.

June: GA of Windows Azure Web Sites Standard and Free tiers.

June: Enhancement to availability, monitoring through Auto Scaling and Alerting Preview for Windows Azure Web Sites, Cloud Services, and Virtual Machines.

June: Dynamics NAV 2013 or Dynamics GP 2013 systems become available for hosting on Windows Azure.

August: GA of Windows Azure Notification Hubs & support for SQL Server AlwaysOn Availability Group Listeners.

Nikhil Sachdeva
Senior Consultant, Microsoft Consulting Services (MCS) - US

Clemens Vasters,
Principal Program Manager, AAPT PM - Germany EMIC

Windows Azure solutions

While in its essence Windows Azure is simply a collection of different kinds of cloud services, it's what you can do with these services that can make the platform appealing to businesses. In other words, it's the solutions that matter most for business customers and not the underlying services needed to implement those solutions.

With almost two dozen different services currently available with Windows Azure, the number and variety of different kinds of solutions that are possible is almost infinite. And that gets us back to our original answer to the question, "What is Windows Azure?" where we stated that:

Windows Azure can be anything you want it to be

But while the sky's the limit as far as what you can do with Windows Azure, Microsoft has identified ten different solution categories where Windows Azure can bring significant benefits to businesses:

- Infrastructure
- Mobile
- Web
- Media
- Integration
- Identity & Access Management
- Big Data
- Dev & Test
- Storage, Backup, & Recovery
- Data Management

Let's briefly examine each solution category in more detail.

Infrastructure

Windows Azure can provide your business with on-demand infrastructure that can scale and adapt to your changing business needs. You can quickly deploy new virtual machines in minutes, and with pay-as-you-go billing you won't be penalized when you need to reconfigure your virtual machines. Windows Azure Virtual Machines even offers you a gallery of preconfigured virtual machine images you can choose from so you can get started as quickly as possible. You can also upload or download your virtual disks, load-balance your virtual machines, and integrate your virtual machines into your on-premises environment using virtual networks.

Mobile

Windows Azure lets you build and deploy a back-end cloud solution for your mobile device apps. You can use popular development platforms like .NET or NodeJS to create your solution, then deploy it to the cloud using Windows Azure Virtual Machines, Cloud Services, or Mobile Services. Windows Azure Mobile Services, in particular, provides cross-platform support for developing solutions for almost any platform including Windows Phone, Windows Store, Android, Apple iOS, and HTML5. Windows Azure Notification Hubs lets you push out notifications to users to enable real-time interactive applications, and you can use social media platforms from Microsoft, Google, Facebook, or Twitter for user authentication purposes.

Web

With support for both SNI and IP-based SSL certificates, and global datacenters with guaranteed SLA and 24/7 support available, Windows Azure can provide you with a robust and secure platform for giving your business website a global presence. You can use Windows Azure Active Directory for authentication and access control, and can securely store your website's business data in Windows Azure SQL Database, NoSQL Tables, BLOB storage. You can create your website using the language of your choice, such as ASP.NET, PHP, Node.js, Python, or even Classic ASP. And for even faster development, you can quickly build your site using a popular framework or template from the Windows Azure App Gallery, which includes WordPress, Umbraco, DotNetNuke, Drupal, Django, CakePHP, and Express.

Media

Windows Azure Media Services makes it easy to give your business a global media presence. You can quickly build end-to-end media workflows using services from both Microsoft and its partners. Your media can be protected using Digital Rights Management (DRM), and Advanced Encryption Standard (AES) or Playready can be used to protect it during playback.

Integration

Windows Azure provides several different options for integrating your existing on-premises infrastructure with your applications running in the Windows Azure public cloud. Windows Azure Service Bus can be used for communicating between your on-premises and cloud-based applications and services. Windows Azure BizTalk Services provides a robust business-to-business (B2B) and application integration PaaS in the cloud. And you can build your integration solution using the familiar tools of .NET and Visual Studio.

Identity & access management

Windows Azure Active Directory (Windows Azure AD) can provide you with identity services running in the cloud that you can use for managing access by employees, partners, and customers to your corporate assets, including both on-premises and cloud assets. You can even synchronize your on-premises Active Directory infrastructure with Windows Azure AD to provide single sign-on (SSO) for users to access your cloud applications, and Windows Azure Multifactor Authentication can be used to provide an additional layer of authentication to help protect your sensitive business data and applications.

Big data

Windows Azure enables you to quickly build a Hadoop cluster based on 100 percent Apache Hadoop. You can then use Windows Azure PowerShell and the Windows Azure Command-Line Interface to seamlessly integrate HDInsight into your existing analysis workflows and gain actionable insights from HDInsight by mining data with Microsoft Excel.

HDInsight supports a wide range of languages including .NET and Java, and .NET developers can also use LINQ and Hive for language-integrated query.

Dev & test

Windows Azure makes rapid application development and testing easy. Instead of having to go through a traditional procurement process and wait for the new hardware you ordered to arrive, you can simply use Windows Azure Virtual Machines to spin up as many virtual machines as you need and perform your application development and testing in the cloud. Then once your application has been validated, you can deploy it into a production environment that's identical to your test environment but also provides you with enhanced performance, infinite scalability, and global reach.

Storage, backup, & recovery

Windows Azure Storage can provide you with secure and reliable storage for all your business needs. Geo-replication across different geographical regions ensures redundancy so you can be sure of being able to access your data in the event that a local disaster occurs. Windows Azure Storage can not only scale to meet whatever needs your business might have, it's also very cost-effective since you only pay for what you use.

Data management

Windows Azure data services can provide you with a consistent experience whether you're working with relational or nonrelational data and currently supports SQL databases up to 150 GB in size. You can utilize your existing data management skills, such as relational database design and Transact-SQL, and can mix and match data across a variety of different data services to create just the solution your business needs.

Windows Azure in action

Microsoft has published numerous case studies of how different companies have utilized Windows Azure to increase their business agility, expand their scale of operations, and reduce their operating costs. Let's briefly look at two examples.

Cooltrax

This Australia-based transportation and logistics company formerly supported its IT needs using its own infrastructure through a local hosting provider. But as their business expanded, they had ever-increasing amounts of tracking data to store and analyze, and they came to the realization that their existing solution couldn't scale in a cost-efficient way to meet their changing needs.

Because Windows Azure provides virtually infinite scalability and supports a diverse set of development technologies, Cooltrax decide to migrate their solution to Windows Azure Cloud Services. They developed their business applications using a combination of Microsoft Silverlight, ASP.NET MVC, and Windows Communication Foundation (WCF). They also used the BLOB service of Windows Azure Data Management to store log files, deployment packages, and backups. And they utilized other Windows Azure services to process incoming data, perform geocoding, generate alerts and notifications, and implement reporting.

As a result of these changes, Cooltrax reduced their IT costs by 15 percent and lowered their hardware maintenance tune by 20 labor hours per month. They now have a single platform that they can utilize using their existing IT expertise.

Open Knowledge Network (OKN)

OKN is an e-learning company based in Madrid, Spain that had been providing SaaS solutions to their customers but was finding that their existing hosting provider was unable to provide them with the level of services they required. They also wanted to expand their business internationally, but their hoster didn't have any datacenters outside Europe.

OKN decided to migrate their e-learning platform for Windows Azure for two reasons. First, Windows Azure supports running cloud applications that are 100 percent open source, and OKN's e-learning platform is based on Linux, PHP, and MySQL. By using Windows Azure Virtual Machines, Cloud Services, and BLOB storage, OKN can run their e-learning platform in Windows Azure. And second, because Windows Azure has datacenters all over the world, OKN can gain a global presence for their e-learning platform so they can grow their business internationally.

> ***MORE INFO*** **You can find these and other Windows Azure mini case studies on the Microsoft Download Center at *http://www.microsoft.com/en-us/download/details.aspx?id=38424*.**

Windows Azure and the Cloud OS

Let's step back for a moment to get the big picture of where Microsoft is going. Windows Azure is only one part of Microsoft's vision for a Cloud OS—an integrated platform that allows businesses to build, deploy, run ,and manage applications on-premises, with third-party hosting services providers, and in Microsoft's public cloud (see Figure 1-2).

The Microsoft platforms that comprise the Cloud OS are actually threefold:

- **Windows Server** A proven, enterprise-class platform that forms the foundation for building cloud solutions.
- **System Center** An integrated platform that provides a common management experience across private, hosted, and public clouds.

- **Windows Azure** An open and flexible cloud platform for building, deploying, and managing applications and workloads hosted on a global network of Microsoftmanaged datacenters.

Windows Server 2012 R2 is the foundation on which businesses, hosters, and Microsoft can build different kinds of cloud solutions. Microsoft System Center 2012 R2 then provides the infrastructure for deploying and managing such cloud solutions. Hosters can use System Center to create public clouds and shared private clouds for their customers. Microsoft uses System Center to create and manage their Windows Azure public cloud platform. And businesses can use System Center to build and deploy on-premises private clouds that can be integrated with cloud services offered by hosters or provided by Windows Azure. Microsoft's Cloud OS thus provides one consistent platform for building, deploying, running, and managing applications on-premises, with a hosting service provider, or in Microsoft's public cloud.

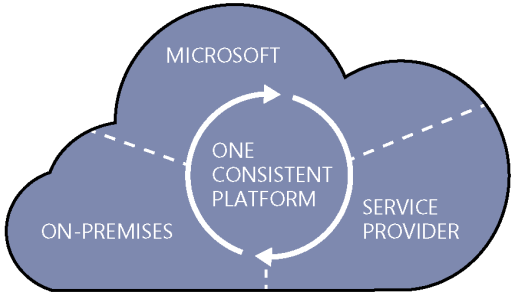


FIGURE 1-2 Microsoft's vision for the Cloud OS.

MORE INFO **For more information on Microsoft's vision for the Cloud OS, see *Introducing Windows Server 2012 R2 RTM Edition* and *Introducing System Center 2012 R2 for IT Professionals*, both of which are available as free downloads from Microsoft Press. And for an example of the Cloud OS at work, see the sidebar titled "Monitoring Windows Azure with System Center Operations Manager 2012 R2" later in this chapter.**

We said earlier that despite the many advantages the cloud can offer, some businesses still resist making use of cloud-based services for reasons relating to concerns about existing IT investments, integration challenges, reliability, control, security, or compliance. By now, however, you probably realize that Windows Azure addresses these different concerns. For example:

- Companies that have made large investments in traditional IT infrastructure don't have to write off those investments. Instead, they can migrate some of their applications and services to Windows Azure while maintaining their existing infrastructure. The different Windows Azure services can then ensure that you can integrate your on-premises and cloud-based environments together so they can function as an integrated solution.

- Enterprises that have a complex, heterogeneous IT landscape who are worried about the challenges they might face if they try to integrate cloud computing into their existing infrastructure can stop worrying. Not only can they selectively choose which portions of their infrastructure to enable for the cloud, they can also utilize the full spectrum of platforms and languages supported by Windows Azure to migrate any portion of their infrastructure that makes business sense to migrate.
- Companies concerned about the reliability of hosting providers can overcome their reluctance to risk moving to the cloud by being armed with knowledge about the architecture of Windows Azure and the top-notch SLAs available to customers. With datacenters around the world, a global CDN, geo-replication, and 24/7 customer support, Windows Azure offers a reliable platform for enabling your business to realize the many benefits of cloud computing.
- While business politics and the traditional in-house IT culture of control can cause a company to resist moving to the cloud, Windows Azure makes it easier for companies to get their feet wet by offering a free trial of Windows Azure. You can find out more about this offering at *http://www.windowsazure.com/en-us/pricing/free-trial/*. In addition, organizations that have Windows Server 2012 R2 and System Center 2012 R2 deployed in their datacenter can try out Windows Azure technologies in-house by downloading and installing the Windows Azure Pack, a collection of Windows Azure technologies available to Microsoft customers at no additional cost that allows you to offer a rich, self-service, multitenant private cloud consistent with the public Windows Azure experience. For more information about the Windows Azure Pack, see *http://www.microsoft.com/en-us/server-cloud/windows-azure-pack.aspx*.
- Finally, organizations that have special requirements in the areas of security and compliance might indeed be blocked from implementing some forms of cloud computing. But they should still keep up with the latest developments with Windows Azure in case the regulatory landscape changes for their industry vertical, and in case some of their business partners, clients, or customers use cloud computing so they can ensure compatibility and flow of information.

Monitoring Windows Azure with System Center Operations Manager 2012 R2

One of the huge features of Windows Azure and its various types of service reduces the amount of management required by clients. There is no need to manage hardware, storage, network infrastructure, and depending on the type of service, not even the operating system. However, this does not mean organizations don't want to be able to view the health of its services running in Windows Azure at various levels.

Operations Manager provides monitoring of the entire organizations IT infrastructure from the hardware all the way through the network, storage, operating system, and applications. The intelligence Operations Manager has about all the different types of components that make up the IT infrastructure is garnered through Management Packs which are imported into the Operations Manager. The exact capabilities of the Windows Azure Management Pack will change over time but at time of writing it has the following capabilities.

- Discovers Windows Azure Cloud Services.
- Provides status of each role instance.
- Collects and monitors performance information per role instance.
- Collects and monitors Windows events per role instance.
- Collects and monitors the .NET Framework trace messages from each role instance.
- Grooms performance, event, and the .NET Framework trace data from Windows Azure storage.
- Changes the number of role instances.
- Discovers Windows Azure Virtual Machines.
- Provides status of each role instance of the virtual machine.
- Discovers Windows Azure Storage.
- Monitors availability and size of each Storage and optionally alerts.
- Discover relationships between discovered Azure resources, to see which other resources a particular Azure resource uses. This information is then displayed in a topology dashboard.
- Monitors management and cloud service certificates and alerts if the certificates are about to expire.
- Includes a new Distributed Application template that lets you create distributed applications that span Azure, as well as on premise resources for Hybrid monitoring scenarios.
- Includes a set of dashboards for the Hybrid monitoring scenarios.

Also remember that if you are using Windows Azure IaaS then the standard Operations Manager agent can run inside those virtual machines along with standard Management Packs to expose even more information if you have a site-to-site VPN to your on premise Operations Manager deployment.

The Windows Azure Management Pack is available from *http://www.microsoft.com/en-us/download/details.aspx?id=38414*. The setup is fairly simple. You will need your Windows Azure subscription ID and a certificate that can be used.

- Import the Management Pack using Administration | Management Packs.
- Under Administration, select Windows Azure and click Add subscription. Enter the Windows Azure subscription ID and specify the certificate file and the password and click Add Subscription

The discovery of the Windows Azure subscription will take some time. Once it's complete we need to configure what elements of Windows Azure should be monitored

Select the Authoring navigation node and select Management Pack Templates and you will see Windows Azure Monitoring. Click Add Monitoring Wizard and select Windows Azure Monitoring which then allows a name, the subscription ID, and then the Windows Azure elements (VMs, storage etc) that should be monitored to be specified, and click Create.

Once the elements are monitored, the information is visible under the Monitoring workspace, Windows Azure

You can also enable various performance counters by modifying the Management Pack Object rules under Authoring Workspace and override the performance counters you care about that aren't enabled. The ability to closely monitor your organizations services running in Azure enables the complete, hybrid view of all your services.

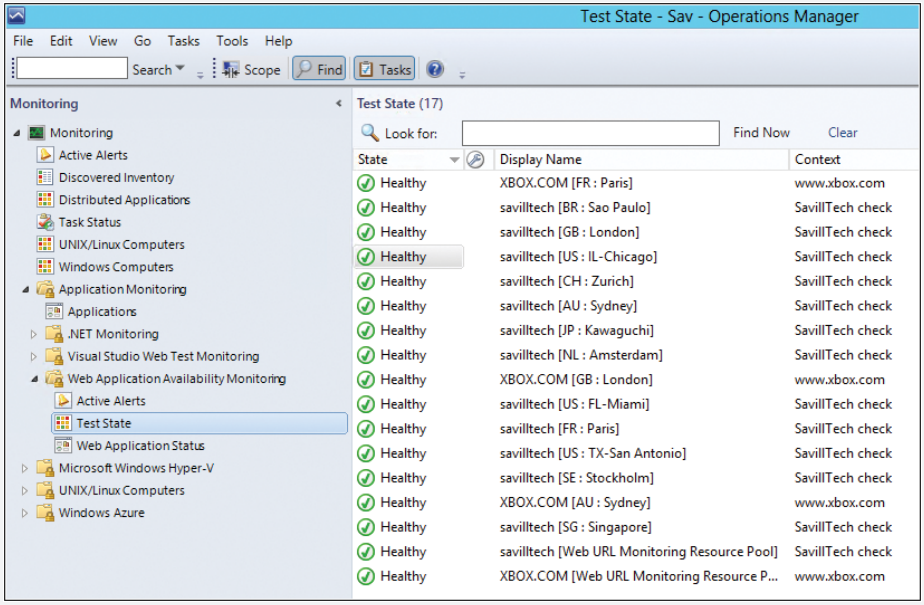


FIGURE 1-3 A view of the health of two websites from all around the world.

There is also a very symbiotic relationship between Operations Manager and Windows Azure. I've talked about how Operations Manager can help monitor your services in Windows Azure, however, it is not a one way relationship. Like many components of System Center, Windows Azure actually brings capabilities to onpremises deployments. Consider Configuration Manager, which can use Windows Azure as an Internet Distribution Point. Data Protection Manager can use Windows Azure storage as backup storage. Operations Manager uses Windows Azure to help it monitor services with an external point-of-view from all around the world. Using the Global Service Monitoring feature the Windows Azure datacenters can be used to periodically perform synthetic transactions against your web services from all around the world and report back on their availability and responsiveness to Operations Manager. In Figure 1-3 I am using some of the Windows Azure datacenters to monitor two websites and I can see they are all healthy.

John Savill
Technical Architect Dallas MTC

Learn more

The following resources available online can help you learn more about what Windows Azure is, how it works, and what it can do for your business:

- For a basic overview of the different components of the Windows Azure platform, see the topic "Introducing Windows Azure" at *http://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/*.
- To start learning today how you can use Windows Azure for building and deploying cloud-based applications, download the August 2013 edition of the Windows Azure Training Kit from the Microsoft Download Center at *http://www.microsoft.com/en-us/download/details.aspx?id=8396*.
- If you are a developer who wants to build and deploy .NET applications on Windows Azure, be sure to check out the Windows Azure .NET Developer Center at *http://www.windowsazure.com/en-us/develop/net/*.
- To install the latest version (2.1) of the Windows Azure SDK for .NET, go to the .NET Developer Center above and click on "install the SDK" under Quick Links. Be sure to read the release notes before installing the SDK. You can also download the Windows Azure SDK for .NET version 2.1 directly from the Microsoft Download Center at *http://www.microsoft.com/en-us/download/details.aspx?id=39708*.
- To keep up with all the latest that's happening with the Windows Azure platform, subscribe to the Windows Azure blog on MSDN at *http://blogs.msdn.com/b/windowsazure/*. Or simply visit *http://www.windowsazure.com* anytime, where you can find everything you want to know about Microsoft's public cloud platform.

CHAPTER 2

Windows Azure compute services

At the core of the Windows Azure platform is its ability to execute applications running in the cloud. Windows Azure currently provides four different models for doing this: Web Sites, Virtual Machines, Cloud Services, and Mobile Services. Together these four approaches comprise the compute services portion of the Windows Azure platform, and they can either be used separately or combined together to build more complex solutions that can meet specific business needs. This chapter describes each of these compute services and provides insights from product team experts concerning how these services relate to one another, what you can do with them, and how you can use them.

Web Sites

Windows Azure Web Sites is a scalable, secure, and flexible platform you can use for building web applications that run your business, extend the reach of your brand, and draw in new customers. It has an easy-to-use self-service portal with a gallery of the world's most popular web solutions including .DotNetNuke, CakePHP, DasBlog, WordPress, and many others. Or you can simply create a new website from scratch and then install a tool like WebMatrix—a free, lightweight web development tool that supports the latest web technologies such as ASP.NET, PHP, HTML5, CSS3, and Node. You can use WebMatrix to create websites and publish applications for Windows Azure. And if you use Microsoft Visual Studio as a development environment, you can download and install a Windows Azure SDK so you can build applications that can take advantage of the scalable cloud computing resources offered by Windows Azure.

When you create a new website you can also have the option to create a database for storing the data for your web application. You can choose to create either a SQL database or a MySQL database for your website. You can also choose to publish your website from source control. This sets up continuous deployment from source control providers like Team Foundation Service, CodePlex, GitHub, or Bitbucket.

Creating a new website

Creating a new website with Windows Azure is so easy we have to show you how to do it. Begin by logging on to the Windows Azure Management Portal at *https://manage.windowsazure.com* using your Microsoft Account username and password. Then select

the Web Sites tab on the left and either click Create A Web Site or click the New button on the command bar at the bottom as shown in Figure 2-1.

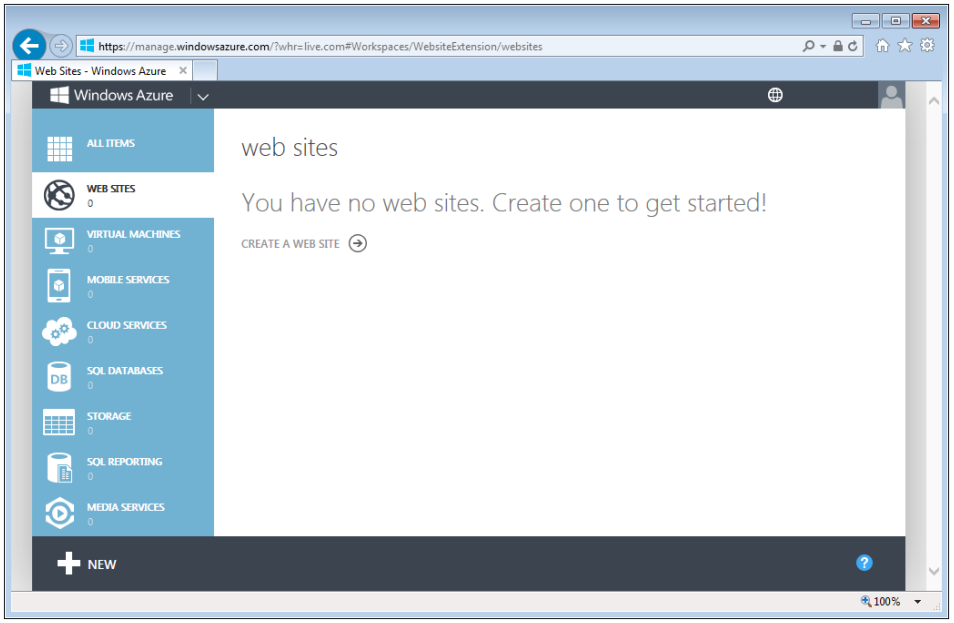


FIGURE 2-1 You may create a new website using Windows Azure.

The command bar then expands, as shown in Figure 2-2, and allows you to quickly create a new website with no additional configuration, a custom website with either a new or existing database, or a new web application based on an application framework, blog engine, template, or any other app available in the Windows Azure Gallery.

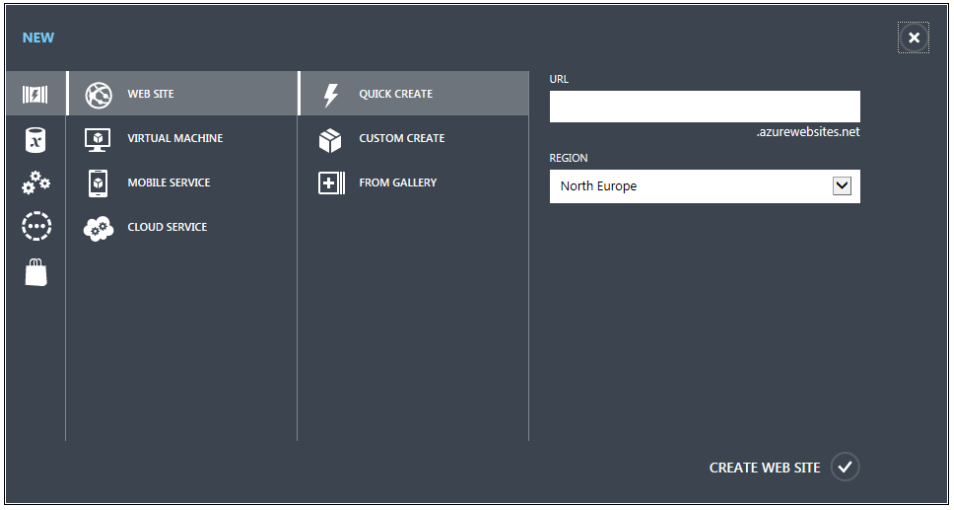


FIGURE 2-2 The Quick Create option.

For example, let's say you want to create a new blog running on WordPress. To do this, simply choose From Gallery when you create your new website, select WordPress from the list of apps available, specify an URL for your new website, choose whether to use an existing MySQL database or create a new one, and specify the region where you want your website to be deployed (for example, North Central US). Then let Windows Azure do the rest, and in minutes your new website will be up and running. Then all you need to do is open your website with your web browser and begin installing and configuring the WordPress application on your website.

Once your web application is installed and running, you can use the Windows Azure Management Portal to monitor and further configure your website as shown in Figure 2-3. The dashboard shown provides you with a quick view of the resources consumed by your website along with information about HTTP server errors should they occur. You can view connection strings, set up deployment credentials, and perform other tasks on your website. You can also browse, stop, restart, or delete your website using the command bar at the bottom of the portal window.

Further monitoring and configuration options are available on the Monitor, Configure, and Scale tabs shown for the website. For example, the Configure tab lets you select the .NET Framework version or PHP version, install SSL certificates, configure SSL bindings, configure custom domains, enable application or site diagnostic logging, configure default documents, and perform other website configuration tasks.

The option for configuring web endpoint status monitoring is new in this release of Windows Azure and is currently in preview status at the time of writing. Preview features represent the latest capabilities that have been added to the Windows Azure cloud platform. These features are in a state of development similar to Release Previews for Microsoft Windows. Preview features have been thoroughly tested and don't cost anything more on top of the standard billing meters needed to access the new feature.

Web endpoint status monitoring allows you to monitor the availability of HTTP or HTTPS endpoints from geo-distributed locations. Using this feature you can test an endpoint from up to three different locations. This is a great way to monitor the global availability of your web applications running in Windows Azure.

Autoscale is another new feature currently in preview in Windows Azure. Windows Azure offers three modes you can use to run your websites: Free, Shared, and Standard. In the Free and Shared modes, all of your websites run in a multitenant environment and have quotas for how much CPU, memory, and network resources they can utilize. The maximum number of sites you can run in Free mode varies with your plan, and the Free mode does not come with an SLA. Free mode is ideal for rapid development and testing because it allows you to test new web application functionality at no cost. Shared mode is currently a preview feature that has less stringent resource usage quotas than Free mode but provides a reduced SLA when compared to the Standard mode.

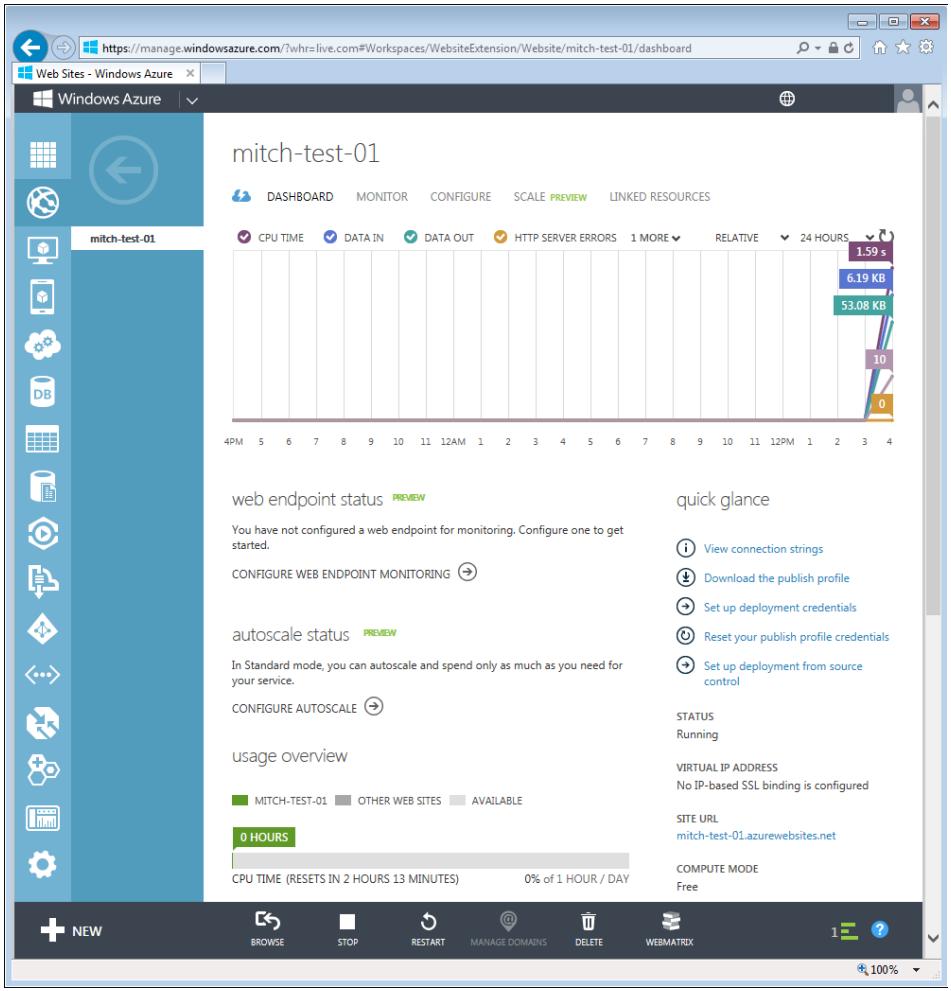


FIGURE 2-3 You can manage and monitor your website in the Management Portal.

When you choose Standard mode, you select which sites run on dedicated per-customer virtual machine instances that allow you to host up to 500 websites. You can manually scale your environment by selecting your instance size from the following:

- Small instance size (1 core, 1.75GB of RAM)
- Medium instance size (2 core, 3.5GB of RAM)
- Large instance (4 cores and 7 GB RAM)

If you enable the Autoscale option for CPU usage, you can configure automatic scale rules based on the activity of your instances. You do this by configuring lower and upper bounds for instance count and target CPU percentage as shown in Figure 2-4.

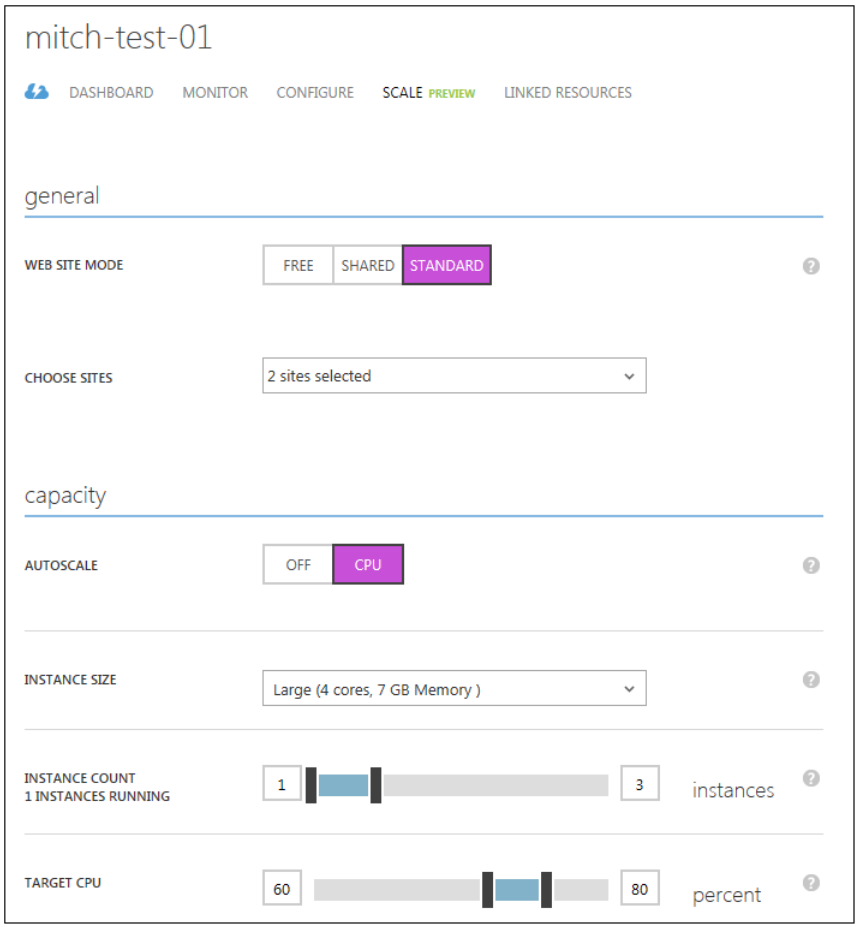


FIGURE 2-4 You use the Scale tab to configure the Autoscale option.

> ***NOTE*** **Autoscale has been added for Web Sites, Cloud Services, and Virtual Machines. Autoscale also supports automatically scaling Mobile Service backends.**

The underlying infrastructure on which Windows Azure Web Sites runs on is Windows Server 2012. This infrastructure is always patched and always running the latest version of frameworks, so you can be sure of having a secure, reliable, and fully functional platform for hosting any business website or web application. In fact, a good way to think of Windows Azure Web Sites is as Platform-as-a-Service (PaaS) for the modern web—a powerful selfservice platform for developers, a flexible hosting solution for IT, and an ideal "white box" solution for hosters. To learn more about Windows Azure Web Sites, check out the links under

"Learn more" at the end of this section. But first let's hear from an insider at Microsoft about the difference between Windows Azure Web Sites and another Windows Azure compute services offering, Windows Azure Cloud Services.

Windows Azure Web Sites vs. Web Role

Web Sites and Cloud Services Web Role are both PaaS offerings in Windows Azure, and they both provide a scalable platform to run your web applications. So a common question is often asked: What is the difference and when should I use which?

To answer these questions, we need to first understand the design intentions of these two different services and the behind-the-scenes implementations. Even though both technologies are PaaS offerings, they provide different levels of abstraction as far as your web application is concerned.

Web Sites

Web Sites has a higher degree of abstraction, because it basically provides Internet information services (IIS) as the platform for you to run your web app. As the result of this, you—the web app developer—do not have direct access to anything below the IIS layer. It still runs on virtual machines, but you are shielded from the operating system by IIS. The virtual machines can be shared by multiple tenants, or they can be dedicated to your websites exclusively. But this shouldn't make any difference to you from a development perspective—all you see is the IIS server.

Because of this design principle, you don't create new virtual machines from scratch every time you request a Web Sites instance, or scale out with additional nodes. Behind the scenes, Windows Azure runs a pool of virtual machines dedicated for the Web Sites service. Each virtual machine is packed with multiple IIS instances (as dense as hundreds of them when appropriate). When you request a site, Windows Azure simply allocates one of the IIS instances in one of the virtual machines to run your app code. If you need to scale, Windows Azure just replicates your assets to additional IIS instances. If you want a dedicated virtual machine instance, Windows Azure takes a virtual machine out of this pool, and only provision your web app onto it.

Because Windows Azure provides the platform at the IIS level, it can achieve much higher management and resource efficiency than if it provided the platform at the virtual machine level, and consequently the cost efficiency can be passed on to you, hence the low unit price for the Web Sites service. In fact, Windows Azure even gives you 10 shared websites for free to get started!

Cloud Services

On the other hand, the Cloud Services Web Role has a lower level of abstraction when it comes to providing a platform of service experience to your web app. Web Role is not fundamentally different from its sibling, Worker Role. The only real difference is that Web Role comes with IIS preinstalled. Every time you provision a Web Role instance, Windows Azure stands up a fresh virtual machine with the Windows Server operating system and IIS installed and configured, dedicated to your role. Windows Azure then deploys your cloud app packages into this virtual machine. This means you, as a developer, have full access to the Windows operating system of the virtual machine during the deployment process as well at the run time. You can run startup scripts to install additional components that run outside of the IIS process (a custom Windows NT service for example), custom-configure IIS, install certificates, open up additional public or internal ports, configure caching services, or even act as the front end of a multitier cloud application inside of a Windows Azure Virtual Network etc., provided that all these tasks can be executed using unattended scripts or through service package configuration. You can also, of course, use Remote Desktop Protocol (RDP) to remote into the provisioned virtual machines, because they are dedicated to your deployment only, although we recommend you do that only for debugging purpose.

Further comparison

Hopefully by now the design principles and the intended consequences are clear to you. You will have more control and more flexibility when choosing Web Role; but you can achieve high cost efficiency (code word for cheaper price) with Windows Azure Web Sites if you do not need the added control and flexibility. So what does this actually mean?

Specifically, Windows Azure Web Sites IIS, as the application server runtime, includes three frameworks at this time: ASP.NET (v3.5 and v4.5), PHP (v5.4 and v5.4), and Node.js. Your ASP.NET, PHP, or Node app can run happily in here. For dedicated instances, you can upload an SSL certificate and you can choose 32-bit or 64-bit IIS mode for your app. You can map a custom domain name to your website as well. Windows Azure even gives you a gallery of some of the most popular open source ASP.NET and PHP website templates including DotNetNuke, CakePHP, Django, WordPress, Orchard CMS, Drupal, etc. You can simply select a template, and Windows Azure will deploy it into your new website to give you a head start. And because of the aforementioned operation model, Windows Azure Web Sites can scale almost instantly because it is not waiting on a new virtual operating system being provisioned. Windows Azure simply allocates additional IIS instances, or virtual machine instances, from an existing pool of such resources. You can expand (or shrink) your web farm size literally in seconds. You can also connect to external

databases through public ports exposed by the database server, but Windows Azure Web Sites cannot be provisioned inside a private Windows Azure Virtual Network. The coolest feature though, is the integration with source control systems such as TFS Service or Git Hub. This enables continuous integration, deployment listing, and roll back—among other things—that any developer will drool over.

With Web Role, IIS is configured with ASP.NET in a fresh instance, but you are not limited to just ASP.NET, you can silent-install any other framework that leverages IIS, or you can install another web server (such as Apache Tomcat) side-by-side with IIS to run your J2EE app. There is really not much you cannot do considering you have the full control of the operating system. So long as Windows Azure Fabric Controller can execute your startup script without human intervention, you can configure the virtual machine until your heart's content. You can also build your Web Role instances as part of a multitiered application under the umbrella of Windows Azure Virtual Network, spanning multiple Cloud Services, including even IaaS virtual machines, with independent scalability at each tier. With all this flexibility though, it comes at a higher cost—and not just the price point. Every time you provision a new Web Role, or scale out an existing Web Role, new virtual machines are created for you from scratch before your app package is automatically deployed and startup script run. This takes minutes compared to the time it takes to scale using Web Sites, which is typically in seconds.

Don't tell anyone but...

Here's a little secret, which shouldn't come as a surprise after the above discussion. Under the covers, Windows Azure Web Sites is actually a giant web application built on top of Cloud Services Web Role instances, in addition to many other components.

Hai Ning
Technology Architect, Microsoft Technology Center

Learn more

For general information about Windows Azure Web Sites and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/web-sites/*.

For more detailed information on what Windows Azure Web Sites is and how it works, see *http://www.windowsazure.com/en-us/documentation/services/web-sites/*.

For a walkthrough on how to create a website using Windows Azure Web Sites, see *http://www.windowsazure.com/en-us/manage/services/web-sites/how-to-create-websites/*.

For more information on Autoscale, see the post titled "Windows Azure July Updates: SQL Database, Traffic Manager, Autoscale, Virtual Machines" on Scott Guthrie's blog at *http://weblogs.asp.net/scottgu/archive/2013/07/23/windows-azure-july-updates-sql-database-traffic-manager-autoscale-virtual-machines.aspx*.

To learn more about web endpoint monitoring, see the post "Windows Azure Portal Update —Configure Web Endpoint Status Monitoring (Preview)" on the Windows Azure Technical Support (WATS) Team Blog at *http://blogs.msdn.com/b/windows_azure_technical_support_wats_team/archive/2013/03/03/minor-update-configure-endpoint-monitoring-preview.aspx*.

Finally, be sure to check out these TechEd 2013 presentations on Windows Azure Web Sites which are available for viewing and download from Channel 9:

- "Windows Azure Web Sites: An Architecture and Technical Deep Dive," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B329*
- "Windows Azure Web Sites and On-Premises Connectivity," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B327*

Virtual Machines

Windows Azure Virtual Machines is a scalable, on-demand IaaS platform you can use to quickly provision and deploy server workloads into the cloud. Once deployed, you can then configure, manage, and monitor those virtual machines, load-balance traffic between them, and connect them to other Windows Azure Cloud Services running web roles and worker roles. You can copy virtual hard disks (VHDs) from your on-premises environment into Windows Azure to use as templates for creating new virtual machines. And you can copy VHDs out of Windows Azure and run them locally in your datacenter.

You can create new virtual machines from a standard image available in the Windows Azure Gallery. Standard images are included for current versions of Windows Server and for different flavors of Linux. Standard images are also available for Microsoft SharePoint, Microsoft SQL Server, and Microsoft BizTalk Server pre-installed on Windows Server. Standard images are a great way of quickly provisioning new virtual machines, but you can also use images you created on-premises to deploy new virtual machines.

> ***NOTE*** **Just added to the Windows Azure Gallery at the time of writing is a standard image of Microsoft Visual Studio Ultimate 2013 Preview. Be sure to check the Gallery often for new virtual machine images.**

As an administrator you have full control over any virtual machines you deploy in the Windows Azure cloud. Virtual machines running Windows Server can be remotely managed using Remote Desktop Protocol (RDP) or with Windows PowerShell. Virtual machines running Linux can be remotely managed using Secure Shell (SSH). Data disks can be created and

attached to virtual machines for storing application data, making it easy to migrate your business applications to the cloud. Additional processor cores can be assigned to virtual machines to optimize performance when running heavy workloads.

Virtual machine instances that you create in Windows Azure are also persistent and selfhealing during hardware failures. This means that any new data created by the virtual machine will not be deleted when the virtual machine is restarted. This is because the virtual disk backing your virtual machine is stored in your Windows Azure storage account. Any data disks you attach to your virtual machine are also persistent and are backed with Windows Azure Storage BLOBs. By contrast, instances of the Web Roles and Worker Roles are self-healing but are not persistent.

Virtual machine charges are calculated by the hour for use, so you can save money by shutting down virtual machines you aren't using. And a monthly SLA that guarantees 99.95 percent uptime means that Windows Azure can be trusted as a reliable way of extending your on-premises workloads to the cloud.

Provisioning a new virtual machine

Creating a new virtual machine in Windows Azure is easy. Just open the Windows Azure Management Portal and select the Virtual Machines tab on the left and click the New button in the command bar at the bottom. The command bar expands and displays two options for creating virtual machines: Quick Create or From Gallery.

The Quick Create option lets you create a new virtual machine which you can configure later. As Figure 2-5 shows, all you need to specify for this option is the DNS name for your virtual machine, the image to use as a template for your virtual machine, the size of the virtual machine (number of cores), a user name and password for administrative access to the virtual machine, and the region or affinity group to which the virtual machine should be assigned.

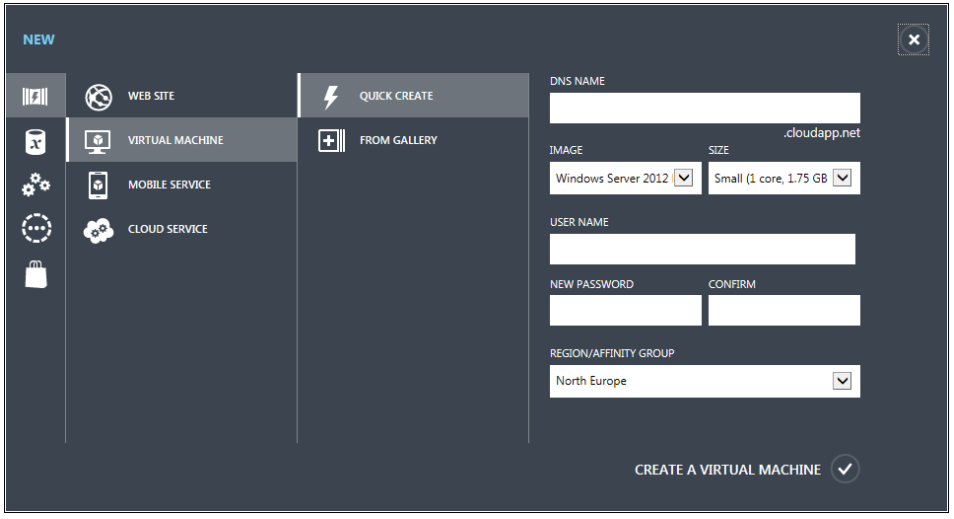


FIGURE 2-5 The Quick Create option for a virtual machine.

The other option, called From Gallery, lets you create a virtual machine by specifying advanced options presented in a series of pages. The first page shown in Figure 2-6 allows you to choose an image to be used as a template when creating your virtual machine.

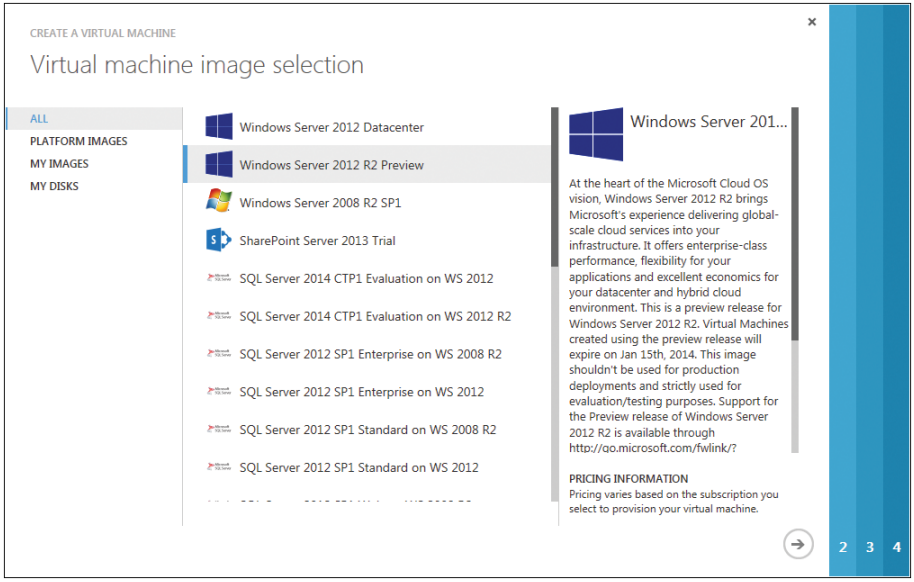


FIGURE 2-6 You can choose an image on which your new virtual machine will be based.

The next page shown in Figure 2-7 is where you specify a hostname for your virtual machine, its size, and username/password for administrative access.

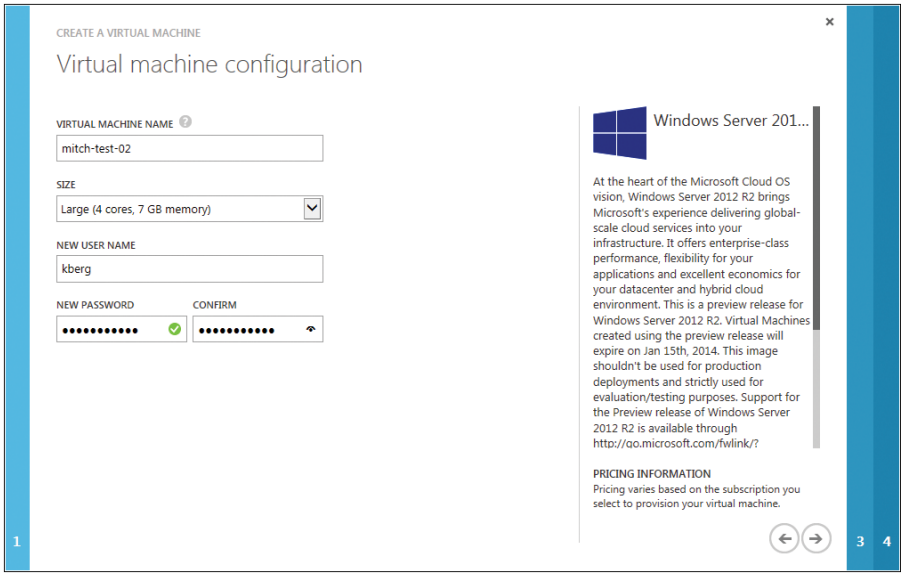


FIGURE 2-7 You continue to configure the new virtual machine on this page.

The remaining pages are where various advanced settings can be configured, for example whether to create a new cloud service or use an existing one; whether to automatically generate a new storage account or use an existing one; the region, affinity group or virtual network to which the virtual machine should be assigned; an availability set to ensure that your virtual machine is not affected by single points of failure, such as the network switch or the power unit of a rack of servers; and the protocols and ports to be used for endpoints like Remote Desktop, Windows PowerShell, or SSH.

Once you've completed the wizard, you can watch it being provisioned using the Windows Azure Management Portal. Figure 2-8 shows one virtual machine running and a second one in the process of being provisioned.

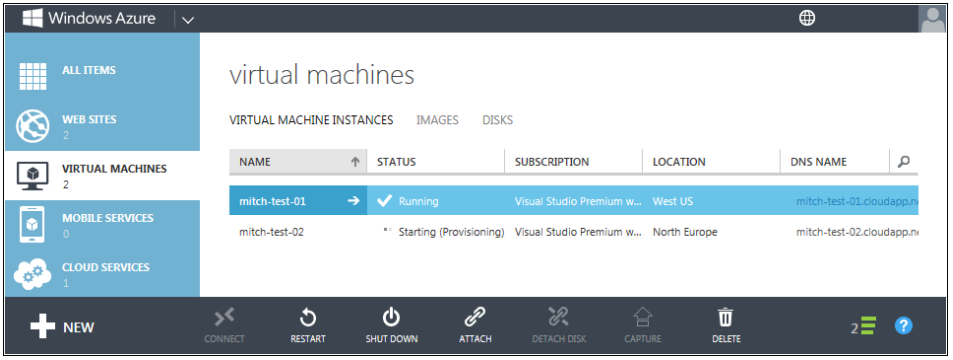


FIGURE 2-8 An example of viewing two virtual machines, one running and the other being provisioned.

Once the second virtual machine is up and running, you can use the command bar at the bottom to perform management tasks like shutting down or restarting the virtual machine, attaching or detaching disks, connecting to the guest operating system of the virtual machine, or deleting the virtual machine.

If you select a virtual machine running some version of Windows Server and click Connect, you are provided an opportunity to download an .rdp file as shown in Figure 2-9.

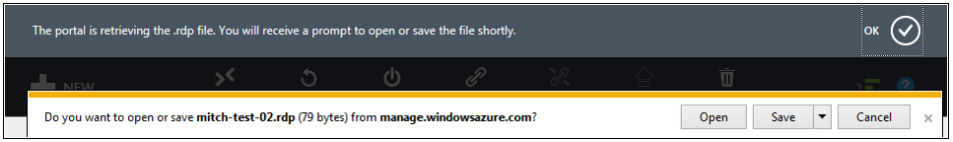


FIGURE 2-9 An example of downloading the .rdp file.

After downloading the .rdp file to your computer, you can double-click it to connect to the virtual machine using Remote Desktop Connection (Mstsc.exe). Connecting to the virtual machine displays its desktop in the Virtual Machine Connect (VMC) window, as shown in Figure 2-10. From here you can further configure the guest operating system of the virtual machine, install and configure applications, and perform other management tasks as if the virtual machine running in the cloud was just another physical server located in your datacenter.

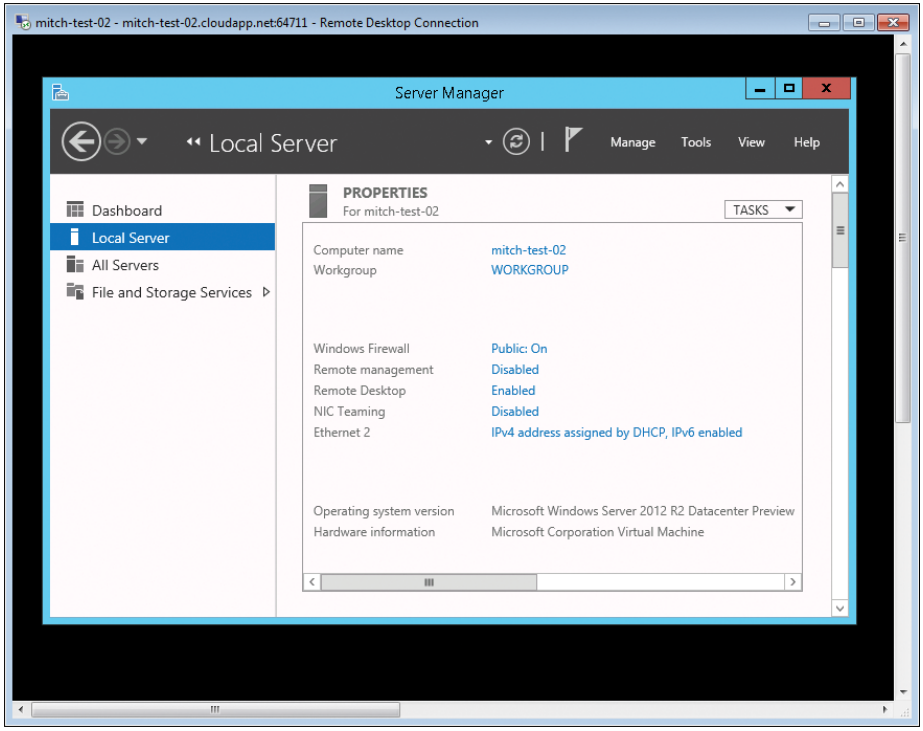


FIGURE 2-10 A new server running in the Windows Azure cloud.

To learn more about Windows Azure Virtual Machines, check out the links under "Learn more" at the end of this section. But first let's learn from an insider at Microsoft about how you can use Windows PowerShell to create and manage virtual machines running in Windows Azure.

Using Windows PowerShell with Windows Azure IaaS

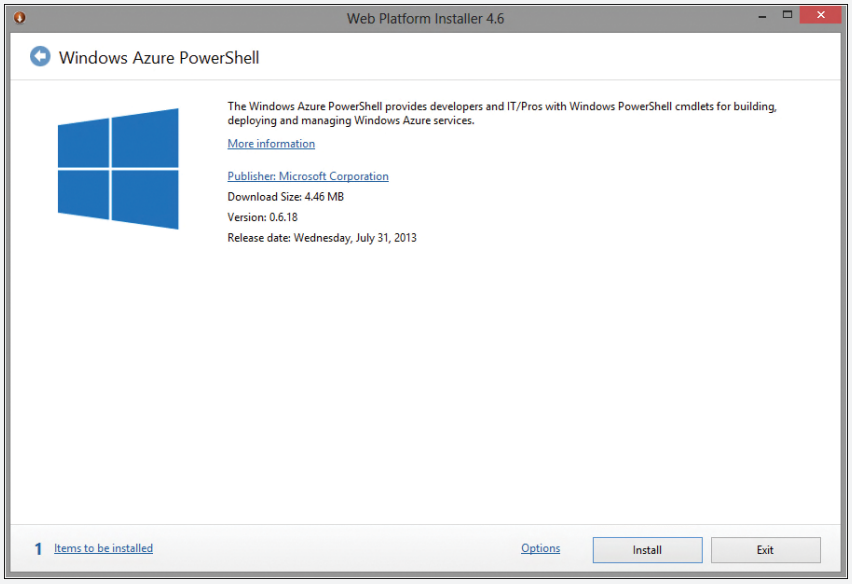
It's very common to focus on the great web interface for Windows Azure IaaS as the means to interact with the environment, create virtual machines, and manage them, however, when thinking about automation, using any web portal is not an option. Microsoft offers a number of automation and orchestration solutions, however, when looking at the strategic direction for automation, workflow, and even command-line interfaces (CLI) for Microsoft technologies your preferred option should be Windows PowerShell. In Windows Server 2012 every aspect of the operating system can be configured and managed using Windows PowerShell. Additionally, not only does nearly every Microsoft offering provide Windows PowerShell cmdlets (essentially commands added to the Windows PowerShell environment) for its management but many third-party software and hardware vendors also offer full Windows PowerShell support and cmdlet modules. As you

would expect, Windows Azure is no exception. It offers a very comprehensive Windows PowerShell module to enable comprehensive Windows PowerShell support.

Windows PowerShell is native to modern versions of Windows with PowerShell v3 in Windows 8 and Windows Server 2012 and PowerShell v4 in Windows 8.1 and Windows Server 2012 R2. The Windows Azure PowerShell module is not provided as part of Windows, however it can be added easily.

Installing Windows Azure PowerShell

The first step is to download the Windows Azure PowerShell module which is available from the Windows Azure Downloads page at *http://www.windowsazure.com/en-us/downloads/?fb=en-us* and can be found at the bottom of the downloads page in the Command Line tools section. Once the Windows Azure PowerShell link is clicked, the Web Platform Installer will launch which will enable you to click Install to complete the Windows Azure PowerShell module installation as shown below. A number of prerequisites are required which will also automatically be downloaded and installed.



The installation of the Windows Azure command tools adds entries with the module loaded that can be found by typing Azure on the Start screen or the module can be imported into existing Windows PowerShell sessions. Depending on your Windows PowerShell environment, configuration may require a change to the execution policy to allow the remotely signed cmdlets to execute. Use the following command to change the execution policy, which must be run from an elevated Windows PowerShell prompt (right-click the Windows PowerShell prompt on search results

and select Run As Administrator from the options which will launch Windows PowerShell with Administrator: at the start of the Windows PowerShell window title) and when prompted press Y:

```
Set-ExecutionPolicy RemoteSigned
```

If the Windows PowerShell environment was not launched via the Windows Azure PowerShell program, then the first step is to actually import the Windows Azure PowerShell module which is accomplished using the following command:

```
Import-Module "C:\Program Files (x86)\Microsoft SDKs\Windows Azure\
PowerShell\Azure\Azure.psd1"
```

Once imported, all the available cmdlets in the Windows Azure module can be viewed using the command:

```
Get-Command –Module Azure
```

Running this command will give you an idea of the full capability of the PowerShell module for Windows Azure as you see nearly 200 cmdlets which are not limited to just Windows Azure IaaS virtual machines but cmdlets specific to web roles, storage, PHP, SQL, services, and much more.

Configuring your environment

Before any actual cmdlets can be used against your Windows Azure subscription, you first have to configure your Windows PowerShell environment to know your Windows Azure subscription and a secure way to communicate with it. There are manual ways to achieve this which are documented at *http://msdn.microsoft.com/en-us/library/windowsazure/jj554332.aspx*. However, I recommend using the Get-AzurePublishSettingsFile cmdlet which will open a browser window that lets you sign-in to your Windows Azure subscription then automatically download the full configuration file. Once this configuration file is downloaded, it is imported using the Import-AzurePublishSettingsFile <file>.publishsettings cmdlet. For example:

```
Get-AzurePublishSettingsFile

<download file from site once entered credentials in web browser and
I saved file in my Downloads folder as WindowsAzureSavillTechMSDN-
credentials.publishsettings>

Import-AzurePublishSettingsFile C:\Users\john\Downloads\
WindowsAzureSavillTechMSDN-credentials.publishsettings
```

Working with Windows Azure PowerShell

At this point, you are ready to use Windows Azure from Windows PowerShell. The first step is to check that you are truly using the Windows Azure subscription you think you are using by running the Get-AzureSubscription cmdlet.

```
PS C:\Users\john> Get-AzureSubscription

SubscriptionName           : Windows Azure MSDN - Visual Studio Ultimate
SubscriptionId             : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Certificate                : [Subject]
                               CN=Windows Azure Tools

                             [Issuer]
                               CN=Windows Azure Tools

                             [Serial Number]
                               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                             [Not Before]
                               8/10/2013 1:29:10 PM

                             [Not After]
                               8/10/2014 1:29:10 PM

                             [Thumbprint]
                               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX

ServiceEndpoint            : https://management.core.windows.net
SqlAzureServiceEndpoint    :
CurrentStorageAccount      :
IsDefault                  : True
CurrentCloudStorageAccount :
```

To view your storage accounts, use the Get-AzureStorageAccount cmdlet—provided you have created a virtual machine before, there will be an implicitly created storage account to use and make a note of the value of the StorageAccountName attribute that will be shown. You need to configure the Windows Azure subscription with a default storage account to make creating a virtual machine easy.

```
Set-AzureSubscription -SubscriptionName "<your subscription name from
Get-AzureSubscription SubscriptionName attribute" -CurrentStorageAccount
"<storage account name from Get-AzureStorageAccount StorageAccountName
attribute>"
```

Creating a virtual machine using Windows Azure PowerShell

At this point, I will assume you have created a virtual machine using the Windows Azure Web Portal and saw the nice list of virtual machine templates available in the Gallery. That Gallery is just scratching the surface of what is really available. Run the command below to see every template that is really available.

```
Get-AzureVMImage | ft Label,ImageName,LogicalSizeInGB
```

I'm going to simply create a new virtual machine using the Windows Server 2012 R2 Preview image and I'm going to create that in the East US datacenter (all locations can be found with the Get-AzureLocation cmdlet). In the code below, I create a virtual machine from the template. The first commands fetch all images and locations to array variables, then when creating the actual VM I use the index number of the actual element I want.

```
$images = Get-AzureVMImage
$locations = Get-AzureLocation
$mySvc = "SavillTechTest"
$myPwd = "P@sswOrd"
New-AzureQuickVM -Windows -name "Sav2012VM" -ImageName $images[25].
ImageName -ServiceName $mySvc -Location $locations[7].Name –AdminUserName
AdminJohn -Password $myPwd
```

The command will take some time, however, it will be visible in the Windows Azure Portal and the cmdlet will show the various actions taken and their successful completion.

```
OperationDescription                                OperationId
OperationStatus                                    --------------------
-----------              ---------------
New-AzureQuickVM - Create Cloud Service             XXXXXXXXXXX
Succeeded                                          New-AzureQuickVM -
Create Deployment with VM Sav... XXXXXXXXXXX                Succeeded
```

You are now managing Windows Azure with Windows PowerShell!

John Savill
Technical Architect Dallas MTC

Learn more

For general information about Windows Azure Virtual Machines and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/virtual-machines/*.

For more detailed information on what Windows Azure Virtual Machines is and how it works, see *http://www.windowsazure.com/en-us/documentation/services/virtual-machines/*.

For a walkthrough on how to create a virtual machine using Windows Azure Virtual Machines, see *http://www.windowsazure.com/en-us/manage/windows/tutorials/virtual-machine-from-gallery/*.

More information on creating virtual machines in Windows Azure can be found on MSDN at *http://msdn.microsoft.com/en-us/library/windowsazure/jj156003.aspx*.

More information on Windows Azure PowerShell can be found at *http://msdn.microsoft.com/library/windowsazure/jj156055*.

Finally, be sure to check out these TechEd 2013 presentations on Windows Azure Virtual Machines which are available for viewing and download from Channel 9:

- "Infrastructure Services on Windows Azure: Virtual Machines and Virtual Networks with Mark Russinovich," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B212*
- "Building Your Lab, Dev, and Test Scenarios in Windows Azure Infrastucture Services (IaaS)," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B370*
- "Best Practices from Real Customers: Deploying to Windows Azure Infrastructure Services (IaaS)," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B361*
- "Crash Course on Automating Deployments in Windows Azure Virtual Machines. How and Which Tools?," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B405*
- "Take Control of the Cloud with the Windows Azure PowerShell Cmdlets," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B305*

Cloud Services

Windows Azure Cloud Services allows you to quickly create, deploy, and manage multitier applications in the cloud. You can define multiple roles for your application to distribute processing and allow flexible scaling of your application. Cloud Services applications can be built using almost any popular development framework including .NET, Node.js, PHP, Java, Python, and Ruby. You can also integrate Windows Azure Mobile Services and Media Services with your cloud application.

With Cloud Services, you can focus your attention on building, testing, deploying, and managing your application instead of focusing on the underlying infrastructure on which your application runs. You don't need to worry about patching servers, dealing with hardware failures, or troubleshooting network issues since Windows Azure is designed to allow applications to be available even in the event of hardware failures or system upgrades.

You can use the Windows Azure Management Portal to monitor the health and availability of applications running on Cloud Services. You can configure alerts so you can be notified in real-time should a service interruption or service degradation occur. And you can use the new Autoscale feature to allow your application to automatically scale up or down as demand changes. This helps minimize the cost of running your applications in the cloud since you only pay for the execution resources that you actually use.

Creating and deploying cloud services

To create a cloud service, you first need to understand a number of concepts. A cloud service role, which consists of your application files and XML configuration files, can be either a web role or a worker role. A web role provides a dedicated IIS web server and is typically used for hosting front-end web applications or mid-tier service layers. Worker roles, on the other hand, host applications that can run asynchronously and are generally used to perform long-running data processing tasks that are independent of user input or interaction.

A role instance is the virtual machine on which the application code and role configuration run. Each role can have one or several instances as defined in the cloud service configuration file (.csdef) which defines the service model for the application. The cloud service configuration file (.cscfg) specifies configuration settings for the cloud service and its individual roles, including the number of role instances. Finally, the service package (.cspkg) contains the actual application code along with the service definition file.

To create a new cloud service in Windows Azure, open the Windows Azure Management Portal, select the Cloud Services tab on the left and click the New button in the command bar at the bottom. The command bar expands and displays two options for creating cloud services: Quick Create or Custom Create. The Quick Create option, shown in Figure 2-11, allows you to quickly create your cloud service by specifying an URL and a region or affinity group.

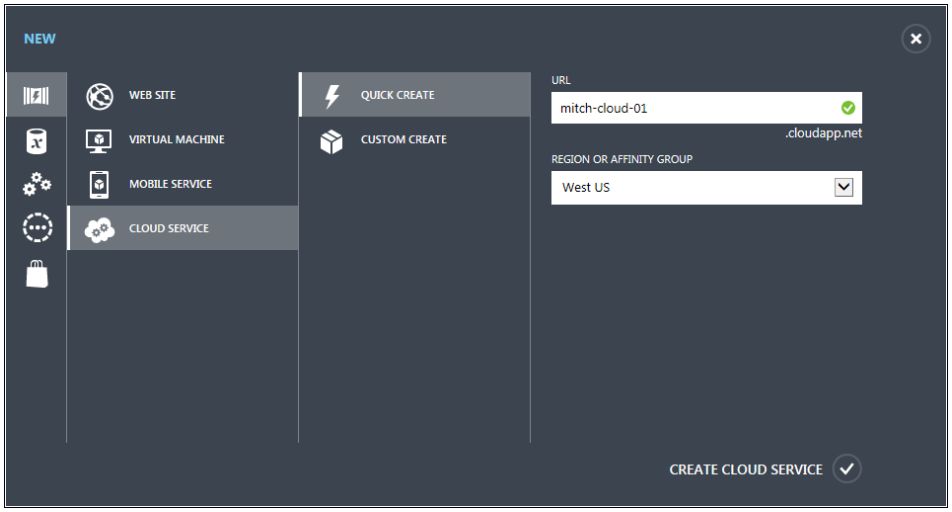


FIGURE 2-11 The Quick Create option is offered to create a cloud service.

Once you've created your cloud service using the Quick Create option, you can install the appropriate Windows Azure SDK needed for running your application and any SSL certificates needed by your application. The next step is to decide which environment you want to deploy your new application to. Windows Azure provides two deployment environments for cloud services: staging and production. The staging environment is where you can test your deployment before you "swap" it into your production environment by switching the virtual IP addresses (VIPs) by which your cloud service is accessed.

Figure 2-12 shows the wizard page you use to create a new staging deployment for your cloud application. You simply specify a name for your new deployment and the location of the service package file (.cspkg) and service configure file (.cscfg). You have the choice of deploying your application even if one or more roles contain a single instance, but you should generally ensure that every role has at least two instances since Windows Azure can only guarantee 99.95 percent uptime if this is the case.

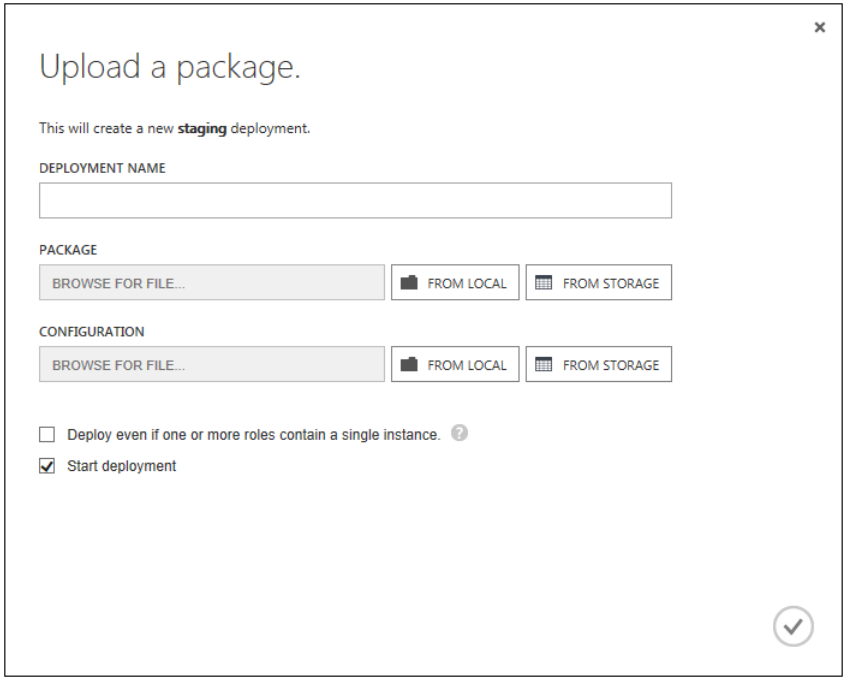


FIGURE 2-12 You upload a package to your cloud service.

The Custom Create option is similar to Quick Create but also enables you to deploy your cloud service package when you create the new cloud service. Just keep in mind these two things:

- Avoid using the Windows registry.
- If you are using a web.config or app.config files, you should instead consider using a service configure (.cscfg) file.

For a more detailed demo of how to create and deploy a cloud service using Windows Azure, see the TechEd 2013 session on this topic under "Learn more" later in this section. Meanwhile, let's hear from another insider at Microsoft about another feature called Windows Azure Diagnostics, which is an application programming interface (API) that allows you to collect diagnostic data from cloud applications running in Windows Azure.

Windows Azure Diagnostics

Running a PaaS application in a Windows Azure Cloud Service provides many advantages. No longer do you need to worry about managing the operating system, but how are you supposed to monitor and troubleshoot?

Windows Azure Diagnostics (WAD) is a framework built into Windows Azure which will facilitate the collection and storing of common logs and data. You can enable WAD within your application or after it has been deployed into Windows Azure.

WAD can be configured to collect the following data from a Windows Azure role instance:

- **Windows Azure logs**
- **IIS logs (web role)**
- **WAD infrastructure logs**
- **IIS failed request logs**
- **Windows event logs**
- **Performance counters**
- **Crash dumps**
- **Custom error logs**

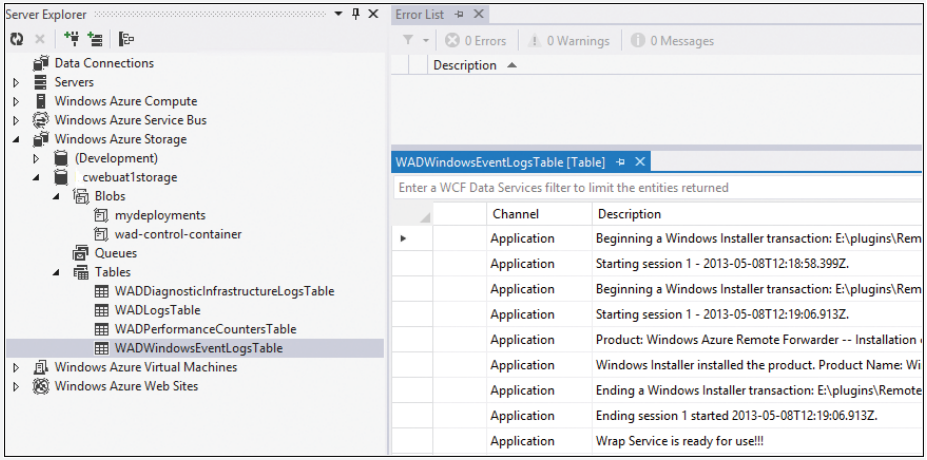
WAD really shines if you are migrating a current application to run on Windows Azure compute services. By utilizing the standard event logs and performance counters, there is no need to replace any existing logging.

WAD will store the data into a specific Windows Azure storage account, I recommend using a dedicated account so access can be segregated from any application data. Depending on what is being collected, it will be stored in either Table or Binary Large Object (BLOB) storage. For example, IIS logs are stored in BLOB storage as files, no different than any on-premises application.

WAD becomes especially important due to the nature of the Windows Azure Fabric. It's critical that logs get pulled off of a Windows Azure instance in case something happens (such as an impromptu reimage). Just as your applications need to be resilient to these situations, so does diagnostics, and the WAD infrastructure will handle it for you.

If you need to edit any settings, you can use Windows PowerShell or one of many third-party tools. You can edit things such as collection interval and how much data to buffer on the instance.

After you have your diagnostics data flowing, you have multiple options to view and consume. Developers can use Visual Studio to view the raw data or copy it locally. For example, viewing event logs of your Hosted Service is easy with Visual Studio, as shown here:



For Operations, we recommend the Cerebrata Azure Management Studio. If you are already using System Center Operations Manager (SCOM) to monitor your service, you will be happy to know that WAD is fully compatible and you can alert and report on data just like your Windows Azure role instance is an on-premises server.

Eric Mattingly
Service Engineer, Microsoft IT – Enterprise Commerce

Learn more

For general information about Windows Azure Cloud Services and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/cloud-services/*.

For more detailed information on what Windows Azure Cloud Services is and how it works, see *http://www.windowsazure.com/en-us/documentation/services/cloud-services/*.

For a walkthrough on how to create and deploy a cloud service using Windows Azure Cloud Services, see *http://www.windowsazure.com/en-us/manage/services/cloud-services/how-to-create-and-deploy-a-cloud-service/*.

For a demo of how to create and deploy a cloud service using Windows Azure, watch the TechEd 2013 presentation titled "Build Your First Cloud App: An Introduction to Windows Azure Cloud Services," which can be found on Channel 9 at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B321*.

To help you decide whether to host your web application using Windows Azure Web Sites, Virtual Machines, or Cloud Services, see *http://www.windowsazure.com/en-us/manage/windows/fundamentals/compute/*.

Mobile Services

Windows Azure Mobile Services is an exciting new feature of Windows Azure that lets you build highly scalable cloud-based backend services for apps running on mobile platforms. The mobile platforms supported by Mobile Services include the Windows Store, Windows Phone, Apple iOS, Android, and HTML/JavaScript applications.

Mobile Services helps you accelerate the process of developing mobile apps by providing you with a turnkey solution for creating new mobile services, allocating backend storage, enabling user authentication, and providing a way to send push notifications to mobile devices. Data for your apps can be stored in Windows Azure SQL Database (see Chapter 4, "Windows Azure data services" for a discussion of this service).

You can use the Windows Azure Management Portal to monitor the health and availability of your mobile services. You can configure alerts so you can be notified in real-time should a service interruption or service degradation occur. And you can use the new Autoscale feature to allow your application to automatically scale up or down as demand changes. This helps minimize the cost of running your mobile services since you only pay for the execution resources that you actually use.

So without any further ado, let's follow along as one of our Microsoft insiders has some fun with Windows Azure Mobile Services.

Having Fun with Windows Azure Mobile Services Scheduler

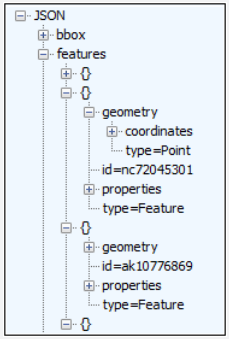
Windows Azure Mobile Services (WAMS) makes it fun and easy to create traditionally complex solutions. Let's have a little fun and create a backend to an application using the WAMS Scheduler that will collect earthquake information from the United States Geological Survey (USGS). With hundreds of earthquakes happening each day (fortunately, most of them are minor) there are a lot of interesting app ideas we can create using this information.

The USGS publishes a specification for their GeoJSON format on their web service page. GeoJSON is a specification designed to make dealing with geo-encoded data more standardized, and JSON (JavaScript Object Notation) has become immensely popular as a lightweight data format. The following URL currently allows you to

get a feed of all earthquakes over the past hour, and other time frames are available as well:

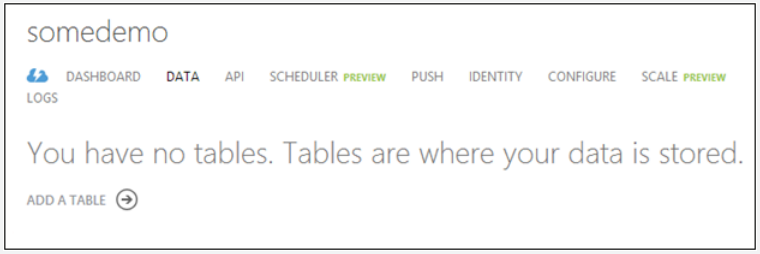
http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson

While you can read about how each earthquake is represented on the USGS web service page, in short, the JSON feed consists of a feature collection, while each earthquake is an item in this collection. Using a tool like Fiddler, we can easily dissect the feed. Fiddler is particularly useful because it can parse and display the JSON in a tree, as shown in the following graphic:



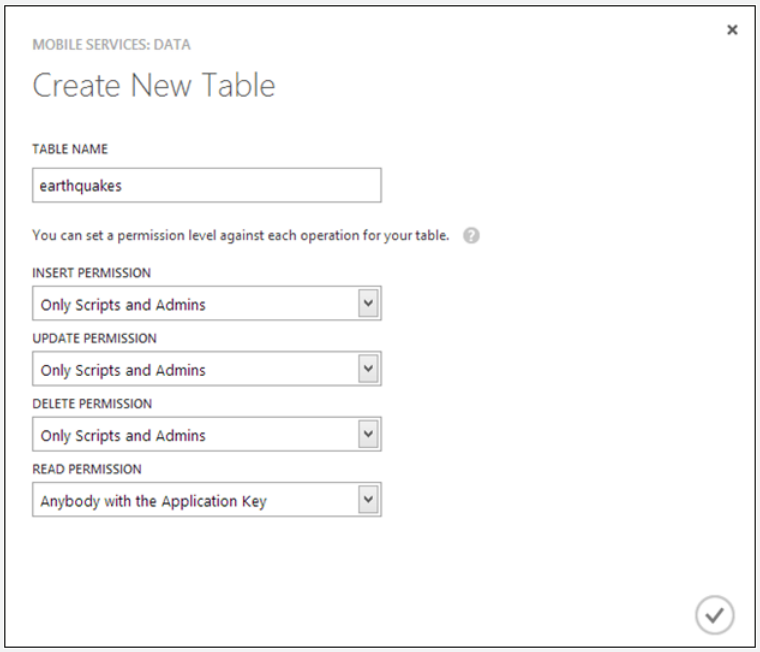
Each earthquake contains both a geometry and a properties object, which contain the data for each earthquake. The geometry is simply an array that contains the longitude, latitude, and depth of the quake, while the properties contain data points about the quake itself, such as the magnitude, time, identifier, etc.

To build our service, we need to regularly consume this feed, parse it, and store it in a database. Then, we need to expose that data so our app can use it. For these kinds of tasks, WAMS really excels and makes building all of this in minutes instead of days. To get started, we'll assume we have an empty WAMS project. We'll add a table to the project by clicking the Data tab, and selecting Add A Table:

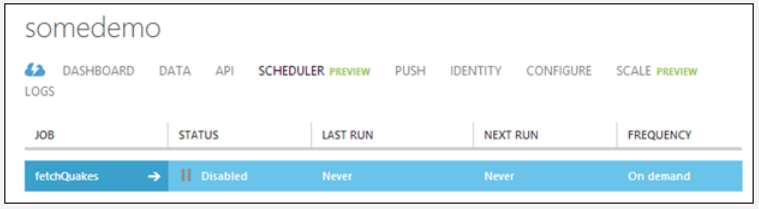


We'll just add a table called "earthquakes." You'll notice that we also have the option to set table permissions. We'll lock it down so that only scripts and admins can change any of the data, and we'll leave read permissions wide open for our app(s) to use. While we're not getting into authentication in this example, this is another

area that WAMS is great at: not only authenticating users, but also being able to integrate that authentication with restricting access to the data.



Next, we'll create a scheduled job to fetch the data at a given interval and store it in the table we just created. We'll call this job "fetchQuakes," and to set it up, we need to click the Scheduler tab and click the Create button near the bottom of the screen. Once set up, we should have an empty job that looks like the following graphic:



Before going further, it's important to understand the way WAMS works with data and scripts. Scripts, at the current time, are all JavaScript based. (In fact, WAMS is an extensible node.js application.) So, everything we write will be written in JavaScript, and we can find reference material by looking at the WAMS Server Side Script page on MSDN.

When storing data, WAMS uses a SQL database. In order to facilitate rapid development, tables have a feature (enabled by default) called dynamic schema. As we consume data and save it to the table, WAMS will add columns automatically and try to predict the correct data type. If it guesses wrong and we'd like to change it (or delete columns), we can connect to the database and make modifications using T-SQL.

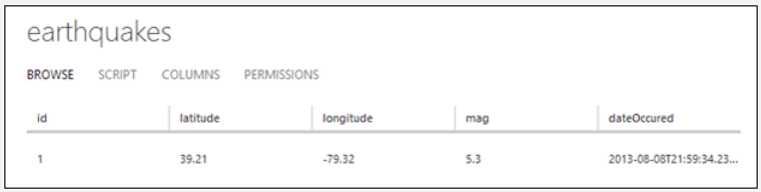
We can edit the job (either the schedule or the script itself) by clicking it. Because it's an empty job, it just has a single method:

```
function fetchQuakes() {
    console.warn("You are running an empty scheduled job. Update the
script for job 'fetchQuakes' or disable the job.");
}
```

JavaScript is a dynamic language so we can simply make up any object we'd like on the fly, and the dynamic schema capabilities should figure out how to store that. For our earthquake example, we'd minimally want to know the latitude, longitude, magnitude, and time of the earthquake. If we put something like the following code in our job and press the Run Once button at the bottom of the screen, we should get a single row in our database, with the columns matching our made-up myQuake object:

```
function fetchQuakes() {
    var myQuake = {latitude: 39.21, longitude: -79.32, mag: 5.3,
dateOccured: new Date()};
    var quakesTable = tables.getTable('earthquakes');
    quakesTable.insert(myQuake);
}
```

In this example, we're using the built-in Tables module to gain access to our database tables, and simply calling insert to insert a new row. We can click the Data tab, select the earthquakes table, and see the data for ourselves:



In addition, if we click the Columns tab, we'll see that it picked a number data type for latitude, longitude, and mag columns, and Date type for the dateOccured column. We might decide at some point to store more properties, and as long as

dynamic schema is on, it modifies the table on the fly. It's recommended that you turn that off (via the Configuration page) before going live.

Now we need to modify the script to consume data from the USGS instead. To do this, we'll tell our script we need to use node's request module to make the HTTP call. To plan this out a little, let's define what our script needs to do: 1) Request the feed from USGS; 2) Iterate each earthquake in the feed; and 3) If the earthquake doesn't exist, insert it.

Our new script looks like so:

```
function fetchQuakes() {
    var httpRequest = require('request');
    var uri = 'http://earthquake.usgs.gov/earthquakes/feed/v0.1/summary/
all_hour.geojson';
    httpRequest(uri, function (err, response, body) {
        if (response.statusCode == 200) {
            var theData = JSON.parse(response.body);
            processData(theData);
        }
    });
}
```

Like most modern JavaScript functions, httpRequest accepts a promise to be invoked when the call is complete. We'll pass the resulting JavaScript into a new method where we'll loop through the result:

```
function processData(json) {
    var quakesTable = tables.getTable('earthquakes');
    var featureList = json.features;
    featureList.forEach(
        function (item) {
            var eq = new earthquake(item);

            quakesTable.where({
                usgsId: eq.usgsId
            }).read({
                success: function (results) {
                    if (results.length <= 0) {
                        //record doesn't exist
                        quakesTable.insert(eq);
```

```
                    }
                }
            });
        }
    );
}
```

Parsing the JSON is easy because JavaScript understands this natively. We can access the root 'features,' and loop through them. There is one more step though—parsing each earthquake. We can abstract that a little inside the foreach, and simply create a new class to do this:

```
function earthquake(item) {
    this.usgsId = item.id;
    this.latitude = item.geometry.coordinates[1];
    this.longitude = item.geometry.coordinates[0];
    this.mag = item.properties.mag;
    this.dateOccured = new Date(parseInt(item.properties.time, 10));
}
```

Not counting whitespace or curly brace lines, this entire script is a mere 26 lines long. Not bad for a script that consumes data from a service and stores it in a database! Of course, this is just a starting point and has tremendous potential.

Want to see this application in use? Check out Earthquake Explorer in the Windows 8 store. It uses this mobile service, and because we're storing the data, it gives us great filtering options. It also sends out tile notifications when new earthquakes are added—another great feature of WAMS!

We could also easily take this cross platform!

Happy coding!

Brian Hitney
Sr. Technical Evangelist

Learn more

For general information about Windows Azure Mobile Services and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/mobile-services/*.

For more detailed information on what Windows Azure Mobile Services is and how it works, see *http://www.windowsazure.com/en-us/develop/mobile/*.

For a tutorial on how to get started using Windows Azure Mobile Services, see *http://www.windowsazure.com/en-us/develop/mobile/tutorials/get-started/*.

Finally, be sure to check out these TechEd 2013 presentations on Windows Azure Virtual Machines available for viewing and download from Channel 9:

- "Build Real-World Modern Apps with Windows Azure Mobile Services on Windows Store, Windows Phone or Android," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B338*.
- "Developing Connected Windows Store Apps with Windows Azure Mobile Service: Overview," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B319*.
- *"Developing iOS and Android Apps with Windows Azure Mobile Services," which can be found at http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B339*.

CHAPTER 3

Windows Azure network services

Windows Azure network services provide the foundation for building hybrid cloud solutions for your business. Windows Azure Virtual Network lets you securely connect your cloud infrastructure to your on-premises datacenter. Windows Azure Traffic Manager allows you to control how user traffic is distributed to cloud services. Windows Azure also includes a name resolution service you can use for internal hostname resolution within a cloud service. This chapter provides an overview of the different network services in Windows Azure and includes insights from product team experts concerning how these services relate to one another, what you can do with them, and how you can use them.

Virtual Network

In the past when you needed to add more servers to your datacenter to handle increasing demand, you had to obtain the necessary hardware and deploy operating systems and applications on your new systems. Such tasks were generally time consuming to perform—hardware procurement can particularly be a bottleneck because of your organization's budgeting process as well as vendor delivery timing.

With the Windows Azure platform, however, you can now easily extend your on-premises datacenter into the cloud, and Windows Azure Virtual Network is key to making this possible. For example, you can use Virtual Network to create and manage a virtual network that uses a private IPv4 address space in Windows Azure. You can also use Virtual Network to create a secure link between your on-premises IT infrastructure and your virtual network in Windows Azure. By creating a hybrid IT infrastructure that combines your on-premises network and your virtual networks in Windows Azure, you can securely connect your cloud-based applications to your on-premises information systems.

Virtual Network can be used with other Windows Azure services such as Virtual Machines. For example, you can use Virtual Network to provide connectivity between virtual machines provisioned using Windows Azure Virtual Machines. This approach is ideal—for example, if you want to run a Microsoft SharePoint farm in Windows Azure. Virtual Network can also be used with Cloud Services. By default, all virtual machines running in the same cloud service can already communicate with each other without

the need for you to create a virtual network for this purpose. By creating additional virtual networks, however, you can also enable virtual machines running in different cloud services to talk to each other.

Creating virtual networks

The best way to become familiar with Virtual Network is to start creating virtual networks. Begin by opening the Windows Azure Management Portal, select the Virtual Network tab on the left and then click the New button in the command bar at the bottom. The command bar expands, as shown in Figure 3-1, and displays the different options available for creating and configuring virtual networks.

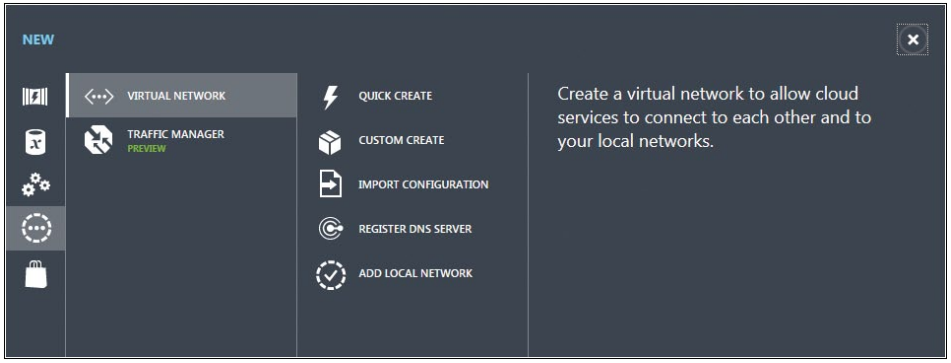


FIGURE 3-1 The command bar offers options for creating virtual networks.

The Quick Create option lets you create a basic virtual network which you can then further configure later on. Custom Create lets you configure your virtual network as you create it, while Import Configuration lets you configure a virtual network by importing an XML configuration file. The remaining options allow you to register DNS servers for your virtual network and add your local network for cross-premises connectivity.

Selecting the Custom Create option launches the Create A Virtual Network wizard. Figure 3-2 shows what the first page of this wizard looks like if this is the first virtual network you are creating. Because virtual networks must be associated with an affinity group, you must create a new affinity group if you haven't done so previously. An affinity group is a logical grouping of Azure services that tells Windows Azure where to locate the services in order to optimize the performance of cloud applications. When you create a new affinity group you must specify the geographical region where your affinity group will reside. For example, Figure 3-2 shows an affinity group being created in the "West US" region. Then if we later create a storage service in the same affinity group, Windows Azure will know that it should locate the storage in the "West US" location in order to optimize performance with other cloud services in the same affinity group.

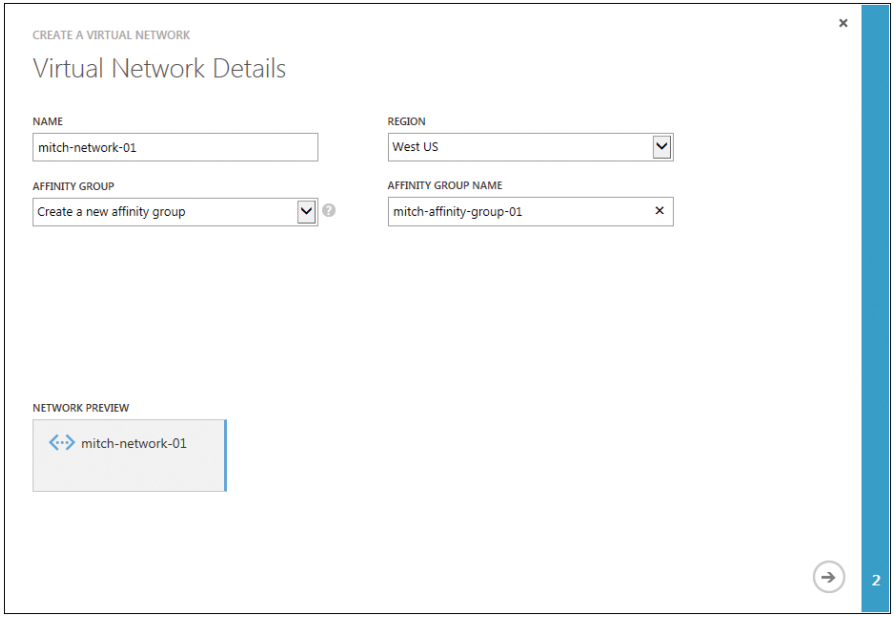


FIGURE 3-2 You can specify the Name and Affinity Group for the virtual network.

The next wizard page. shown in Figure 3-3, allows you to add DNS servers to your virtual network for name resolution. See the sidebar titled "Windows Azure and DNS name resolution" later in this section for more information on the different ways you can configure DNS name resolution for Windows Azure.

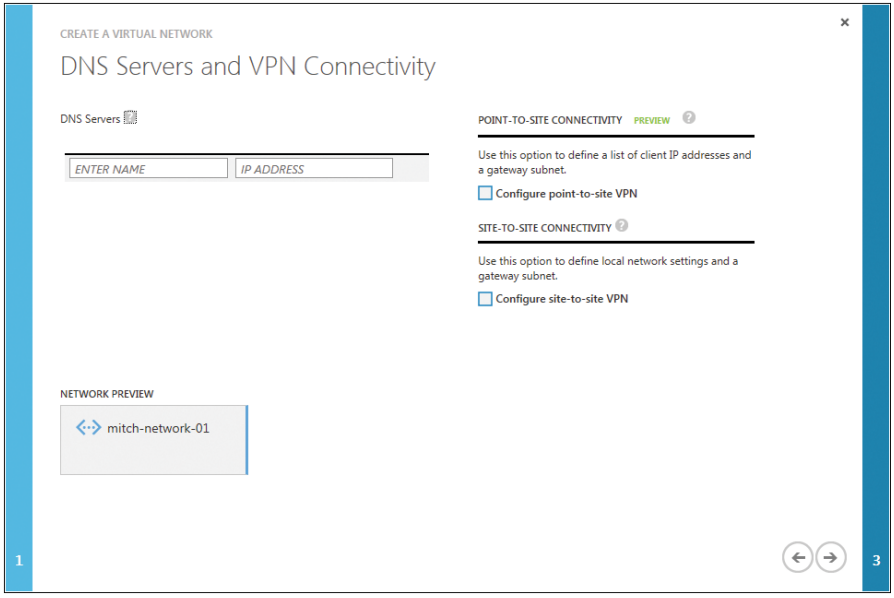


FIGURE 3-3 You can specify DNS servers and VPN connectivity options.

You can also use the wizard page shown in Figure 3-3 for configuring two kinds of VPN connectivity:

- **Point-To-Site Connectivity** Selecting this option displays an additional wizard page where you can define the address space for VPN clients that will connect to your virtual network from outside your virtual network space. This feature is currently in preview at the time of writing.
- **Site-To-Site Connectivity** Selecting this option displays an additional wizard page where you can define the VPN device IP address and address space used by clients connecting your virtual network from a remote site. You might use this option, for example, to establish a secure site-to-site VPN connection between your on-premises corporate network and your virtual network in the cloud. For more information on this feature and how to configure it, see the expert sidebar titled "The secure door to hybrid IT—Windows Azure site-to-site VPN" from a Microsoft insider later in this section.

If you haven't selected either of the two options above, the next and final wizard page shown in Figure 3-4 is where you can add address spaces and subnets to your virtual network. By default, the address space 10.0.0.0/8 is automatically added and the subnet 10.0.0.0/11 created, but you can edit both of these to change your virtual network addressing if desired, for example by changing the address space to 172.16.0.0/12 or 192.168.0.0/16 or something different. Only private IP address ranges can be specified as address spaces however.

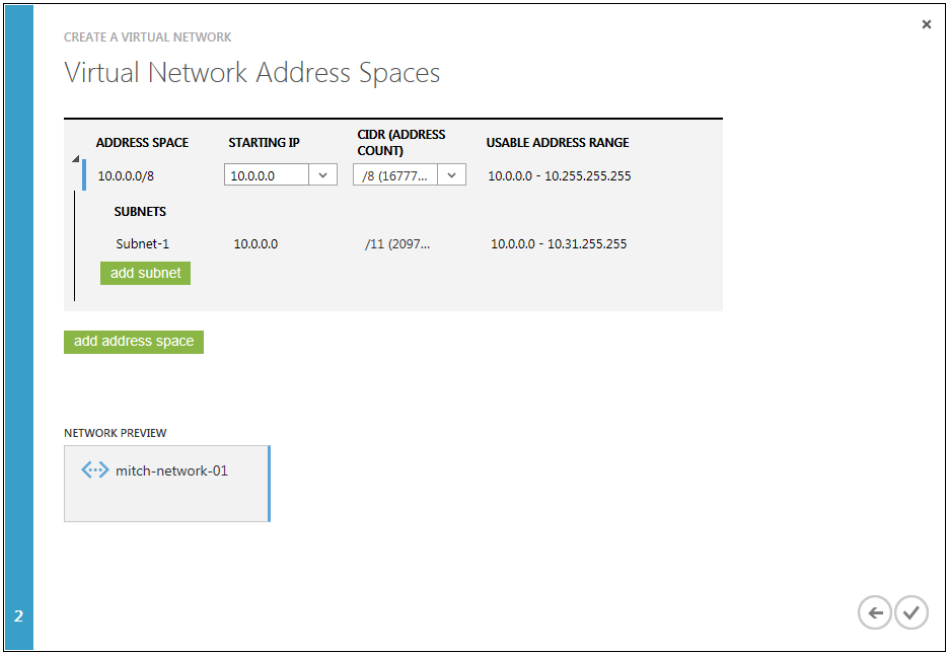


FIGURE 3-4 You can specify address spaces and subnets.

Windows Azure and DNS name resolution

Before you begin deploying virtual machines or role instances in Windows Azure, you need to consider and plan for how the DNS names of these virtual machines will be resolved from your on-premises network. Windows Azure provides its own name resolution service that can be used for resolving instance names within the same cloud service. For example, if you have two virtual machine instances named SRV-A and SRV-B running within the same cloud service named CLOUD-C, you don't need to deploy and configure a DNS server in order for each server to resolve the fully-qualified domain name of the other server.

If, however, your virtual machine instances are running in separate cloud services, Windows Azure name resolution won't suffice. You'll need to use a DNS solution of your choice for such purposes, for example a public DNS server, a DNS server belonging to your Internet service provider, or a DNS server on your corporate network.

For a comprehensive list of the different name resolution scenarios possible for Windows Azure and the solutions you can choose from, see *http://msdn.microsoft.com/en-us/library/windowsazure/jj156088.aspx*.

Once you've created one or more virtual networks in Windows Azure, you can associate a virtual network subnet with a virtual machine when you create the virtual machine using the procedure shown previously in Chapter 2, "Windows Azure compute services." For example, Figure 3-5 shows a new virtual machine being created and associated with a previously created cloud service named mitch-test-all and with the Subnet-1 (10.0.0.0/11) virtual network subnet.

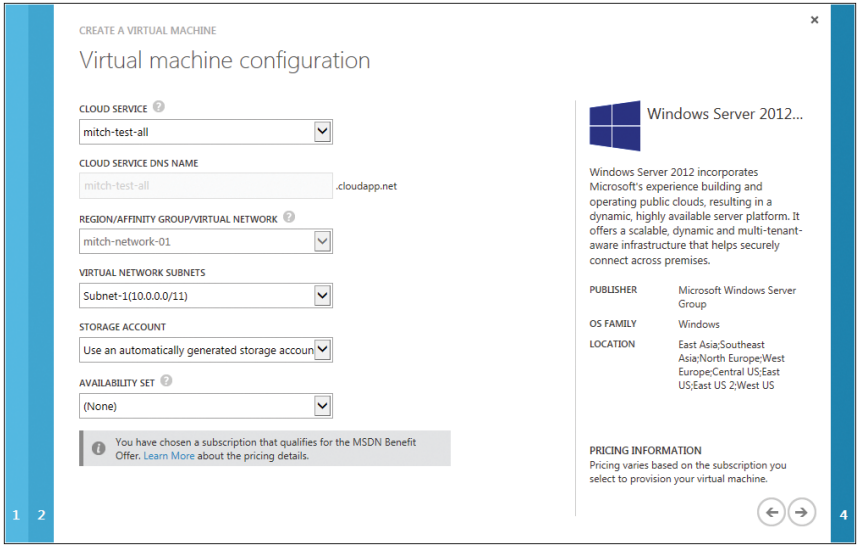


FIGURE 3-5 You can specify a virtual network to use when creating a new virtual machine.

Creating multiple virtual machines associated with the same cloud service and same virtual network subnet is a great way to quickly create a test network of servers in Windows Azure. Figure 3-6 shows three virtual machines associated with the mitch-test-all cloud service and the Subnet-1 virtual network subnet. To display this information, we first selected the Cloud Service tab on the left of the Management Portal. Next, we selected mitch-test-all from the list of cloud services displayed. Finally, we selected the Instances tab to show all virtual machine instances running in this cloud service.

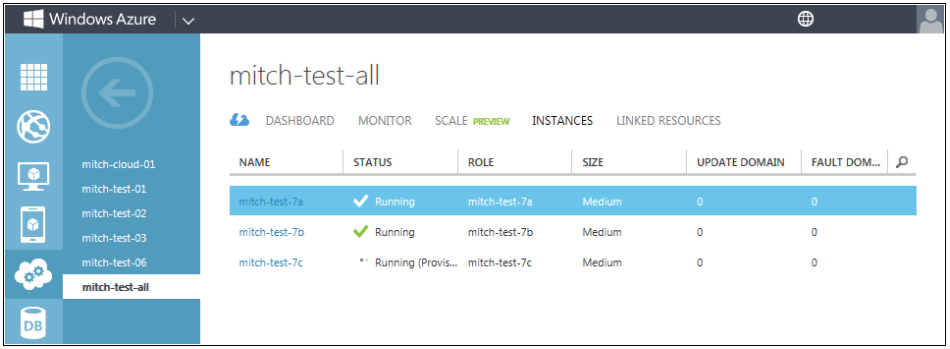


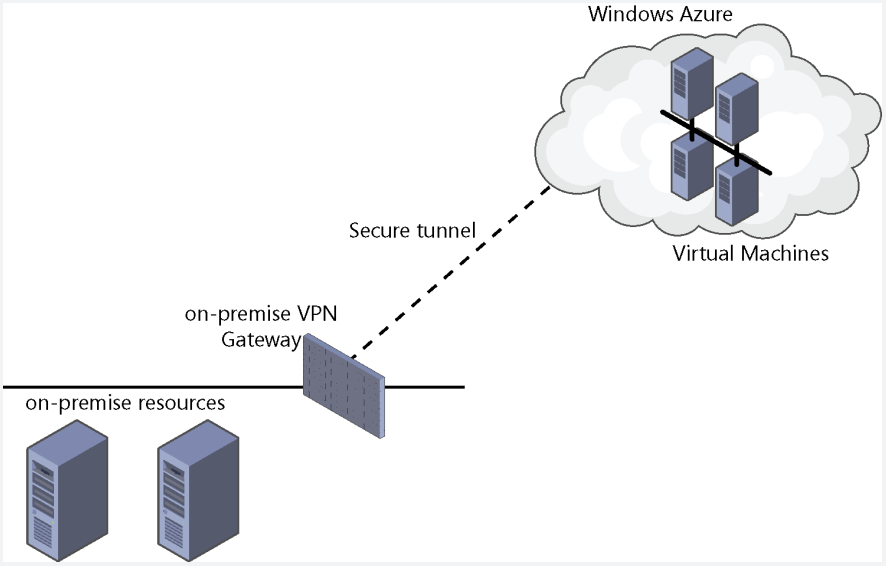
FIGURE 3-6 Three virtual machine instances are running in the same cloud service and on the same virtual network.

If we then selected each virtual machine in turn and displayed its dashboard page, we would see that Windows Azure has assigned the first three available IP addresses in Subnet-1 to the virtual machines, namely the addresses 10.0.0.4, 10.0.0.5, and 10.0.0.6.

The Secure door to hybrid IT—Windows Azure site-to-site VPN

Many companies have already realized the potential that cloud computing has to offer. However, some of these companies opted to invest more in building their own private cloud infrastructure in house. Many of these companies moved to this solution primarily because there were security concerns regarding moving their entire data to a public cloud provider. While private cloud is certainly a very good approach to take full advantage of the essential characteristics of cloud computing, there is more on the horizon for companies that are looking for agility, security, and elasticity. It is exactly on this context that hybrid cloud is turning into a great investment. Companies can now decide what data goes to a public cloud provider and what stays on-premises. What are the scenarios where the return on investment to have VMs in the cloud is better than having them on-premises? Hybrid cloud enables companies to have the best of both worlds: cloud computing and on-premises resources.

With Windows Azure IaaS, companies do not have the capability of instantiating VMs on Windows Azure and securely accessing these VMs from their resources onpremises. This is done using a site-to-site VPN between the on-premises VPN gateway and Windows Azure gateway, as shown in the following diagram:



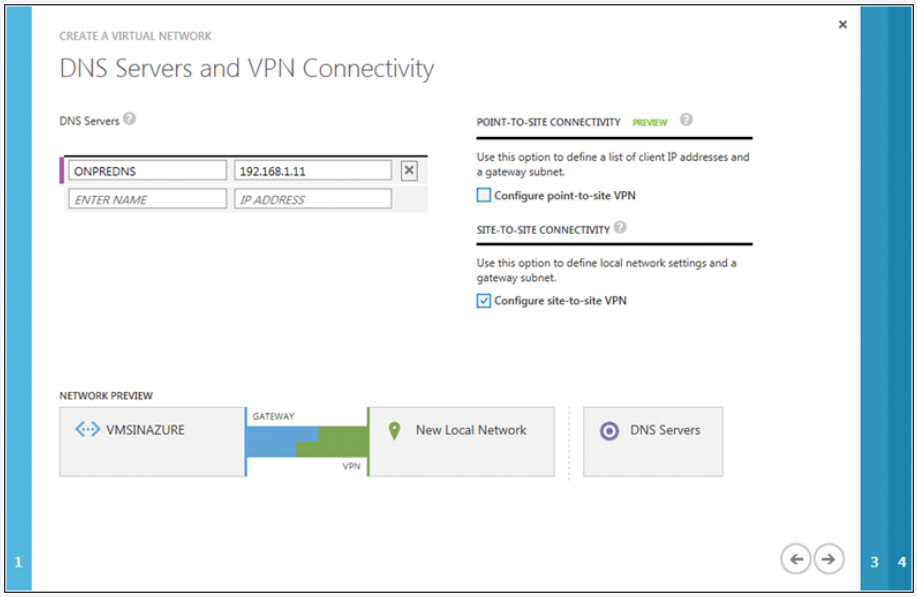
In order to set up a site-to-site VPN with Windows Azure gateway, the on-premises VPN device must support IKE v1 or IKE v2. Keep in mind that if you use IKE v1, Windows Azure will only support static routing. IKE v2 which uses dynamic routing is currently in preview at the time of writing. For a list of supported VPN devices and settings that must be used, see *http://msdn.microsoft.com/en-us/library/windowsazure/jj156075.aspx*.

This setup enables companies to rapidly and securely deploy VMs in Windows Azure. There are many scenarios that this setup can be used for such as development and testing. You can setup VMs in Windows Azure to be accessed from on-premises workstations to validate and test applications. The rapid instantiation of resources can assist you with quickly validating applications before they go to production.

Companies can also take advantage of the following capabilities in Windows Azure in order to make the solution highly available and enhance the overall security:

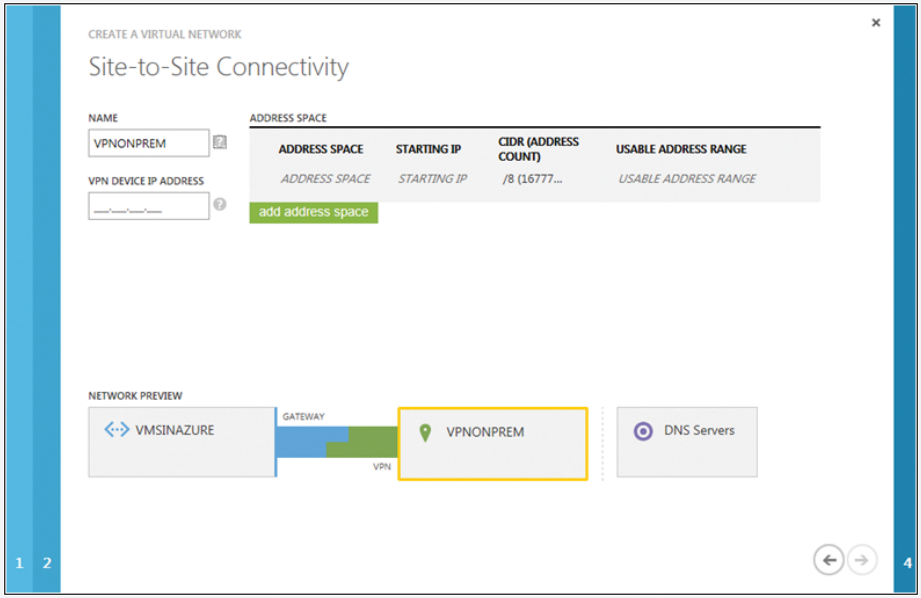
- Windows Azure Load Balancing
- Windows Azure Active Directory
- Windows Azure Autoscale

In order to implement the first step of this setup you will need to create a virtual network to host the VMs and the Windows Azure gateway. When you launch the Create A Virtual Network wizard, you will have the option to select Site-To-Site Connectivity, as shown here:



On this page you can also specify the DNS Server that will be used by the VMs that are instantiated on this virtual network. Since you want to have connectivity with the on-premises resources, you should type the DNS Server name and IP Address for the DNS located on-premises. The next step is to specify the On-Premises VPN Device name and IP address, as shown here:

It is very important that your on-premises device has the configuration and it is on the supported list of devices. For a list of supported VPN devices and settings that must be used, see *http://msdn.microsoft.com/en-us/library/windowsazure/jj156075.aspx*. Once you finish this setup, the Windows Azure gateway will be created and the public IP address of the device will be provided to you. At this point, you should start the configuration on your on-premises VPN device.



For a comprehensive scenario, considerations, and implementation of a hybrid IT solution, read the "Hybrid IT Infrastructure Solution for Enterprise IT" article set at *http://aka.ms/hybriditinfrastructuresolution*. In this document set, produced by our team (*http://technet.microsoft.com/en-us/cloud/private-cloud*), we describe in detail how to plan, design, and implement a solution that leverages all those capabilities and more.

Yuri Diogenes
Sr. Technical Writer

Learn more

For general information about Windows Azure Virtual Network and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/virtual-network/*.

For more detailed information on what Windows Azure Virtual Network is and for tutorials on how to create and configure different kinds of virtual networks, see *http://www.windowsazure.com/en-us/documentation/services/virtual-network/*.

Additional documentation on Windows Azure Virtual Network can be found on MSDN at *http://msdn.microsoft.com/en-us/library/windowsazure/jj156007.aspx*.

Traffic Manager

Traffic Manager is another network service available with Windows Azure. Traffic Manager lets you load balance incoming traffic across multiple hosted Windows Azure services regardless of whether they're running in the same datacenter or in different ones at different geographical locations around the world. By using Traffic Manager, you can distribute users to the "best" location in your Windows Azure cloud solution to ensure high performance, availability, and resiliency for your cloud-based applications.

Traffic Manager is currently in preview at the time of writing and is free of charge.

Using Traffic Manager

Traffic Manager works by applying an intelligent policy engine to DNS queries for your domain names. To use Traffic Manager, you simply create configurations using the Management Portal. A Traffic Manager configuration consists of a profile, a definition, a policy, and monitors. A profile contains a domain name prefix that you create and is visible in the Management Portal. The definition contains the policy settings and monitor settings for the profile. Policy is where load-balancing methods and endpoints are specified. And monitors are where the DNS timeout, protocol, port, and relative path are specified. The process by which Traffic Manager routes traffic is explained in detail at *http://msdn.microsoft.com/en-us/library/windowsazure/hh744833.aspx*.

Traffic Manager provides you with a choice of three load-balancing methods: performance, failover, or round-robin:

- **Performance** This method directs traffic to the closest service based on network latency.
- **Round-robin** This method distributes traffic equally across all services.
- **Failover** This method directs traffic to the backup service if the primary service fails.

As Figure 3-7 shows, you can select the load-balancing method you want to use when you use the Quick Create option to create a new Traffic Manager profile using the Management Portal.

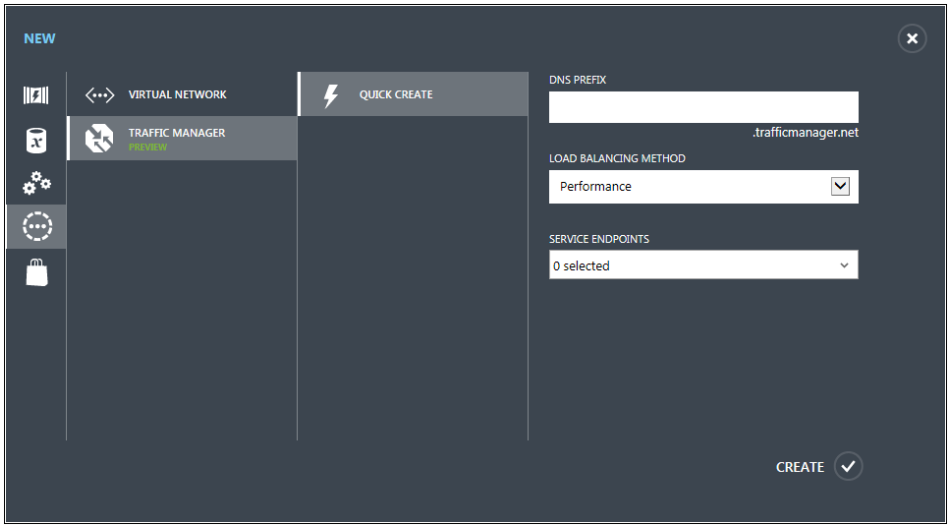


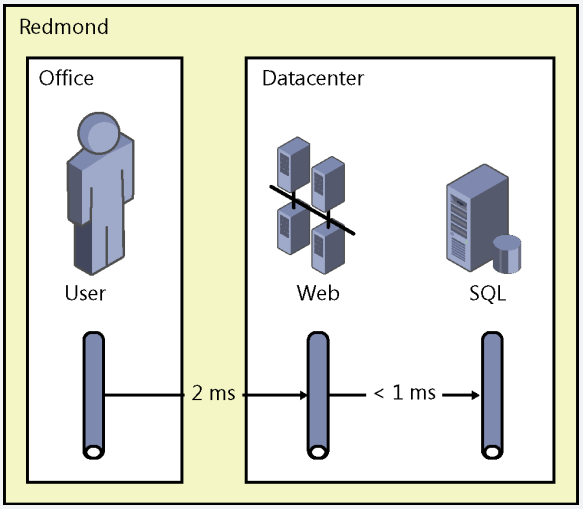
FIGURE 3-7 You can choose the load-balancing method when creating a new Traffic Manager profile.

For further information on how to plan and implement Traffic Manager for your cloudbased applications, see the links under "Learn more" at the end of this section. But first let's here from one of our insiders at Microsoft on why you need to consider network latency when planning your cloud-based applications.

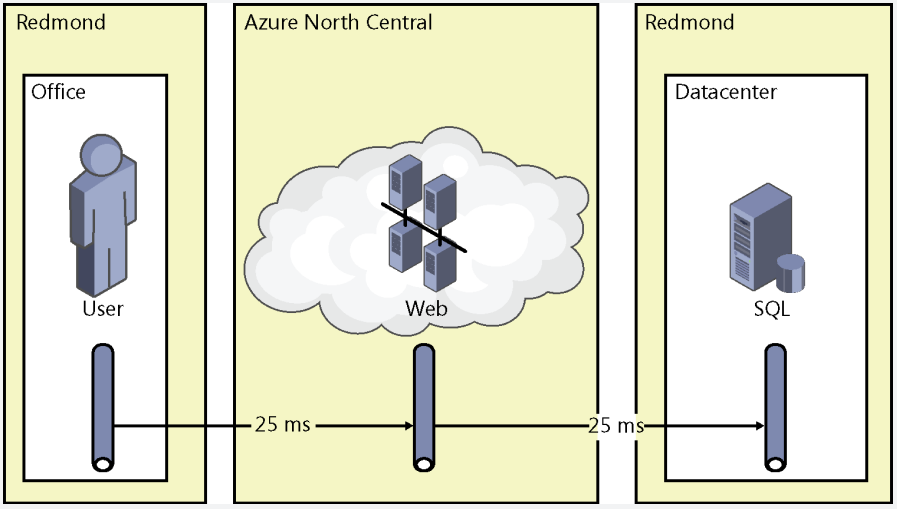
Impact of network latency on hybrid applications

When building hybrid applications, either by migrating an existing app or building net-new, it's important to understand the impact that network latency may have. When Microsoft IT began to explore migrating applications to a hybrid model, we realized quickly that our applications might not respond perfectly to going hybrid.

Take a classic two-tier application. In its original topology the users, web servers, and SQL servers were all within a few milliseconds of each other.



We wanted to move just the web servers into Windows Azure and keep SQL onpremises. By doing this, we suddenly introduced over 50 milliseconds of latency into a topology which previously was under 5 milliseconds! Needless to say, the application was never tested under these conditions....



How did our application react? Not well. We experienced a range of performance problems from general slowness to timeouts. Typically, the worst impact was seen when we were performing a high number of operations while passing large amounts of data.

We learned that the key is to be "chunky" instead of "chatty" over high-latency connections. Sometimes this may mean only a few changes while other times it

might mean a redesign. In our situation, we analyzed all the calls to SQL and noticed hundreds of redundant calls for a single page click due to Entity Framework. When fully on-premises, this was never noticed as the performance was still acceptable and the SQL calls took milliseconds.

Latency will also potentially impact on the "last mile" as well. Our application users sat within 1 millisecond of the datacenter but after migrating the web front ends to Windows Azure that changed to 25 milliseconds. This is thankfully more commonly tested as you are unable to completely control a user's latency to your application.

Now if it's a new hybrid application, we design with latency in mind. We try to use both Windows Azure Cache and the CDN to get data as close to users or the servers as possible. If migrating, we test the application using software network emulators beforehand so we know upfront what the experience may be.

While it's impossible to say exactly how latency might affect your application, it's something to look out for.

Eric Mattingly
Service Engineer, Microsoft IT – Enterprise Commerce

Learn more

For general information about Windows Azure Traffic Manager and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/traffic-manager/*.

For more detailed information on what Windows Azure Traffic Manager is, how it works, and how to plan and implement its use, see *http://www.windowsazure.com/en-us/documentation/services/traffic-manager/*.

For a short walkthrough of configuring Traffic Manager to help make a cloud service highly available and reliable, see the post titled "Windows Azure July Updates: SQL Database, Traffic Manager, Autoscale, Virtual Machines" in Scott Guthrie's blog at *http://weblogs.asp.net/scottgu/archive/2013/07/23/windows-azure-july-updates-sql-database-traffic-manager-autoscale-virtual-machines.aspx*.

For a demonstration of Traffic Manager, see the Microsoft TechEd 2012 presentations titled "Overview of New Networking Features in Windows Azure," which is available for viewing and download from Channel 9 at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2012/AZR304*.

CHAPTER 4

Windows Azure data services

Windows Azure data services allow businesses to store, access, analyze, and protect their data while making it available from anywhere and at any time. From storing data in SQL databases in the cloud to analysis and reporting, Windows Azure provides different services that can meet the needs of your business. Whether your data is stored on-premises or in the cloud, Windows Azure helps keep your data safe with automated backups. Windows Azure also ensures high throughput and low latency with managed caching of application data. And it can help protect private cloud solutions deployed with Microsoft System Center 2012 by replication and recovery at a secondary location. This chapter provides an overview of the different data services in Windows Azure and includes insights from product team experts concerning how these services relate to one another, what you can do with them, and how you can use them.

Data Management

Microsoft SQL Server is widely used as a modern data platform for solutions that help businesses unlock the hidden value in their data. For businesses that rely on the SQL Server database model, Windows Azure offers several ways of extending their database solutions to the cloud. Let's take a look at these now.

SQL Server in Windows Azure Virtual Machines

Windows Azure Virtual Machines can be used to run your existing SQL Server applications in Windows Azure. This approach might be best if your applications require full SQL Server functionality and your IT staff can handle the job of maintaining and patching the underlying Windows Server guest operating system and SQL Server database instances. It also makes for fast and easy migration of SQL databases from on-premises to the cloud with no code changes required.

Deploying a new virtual machine with SQL Server installed is easy because the Gallery includes a wide range of images you can choose from when creating your virtual machines (see Figure 4-1). This ease and simplicity makes this approach ideal for development and test work since you can spin up new virtual machines running SQL Server anytime you need them.

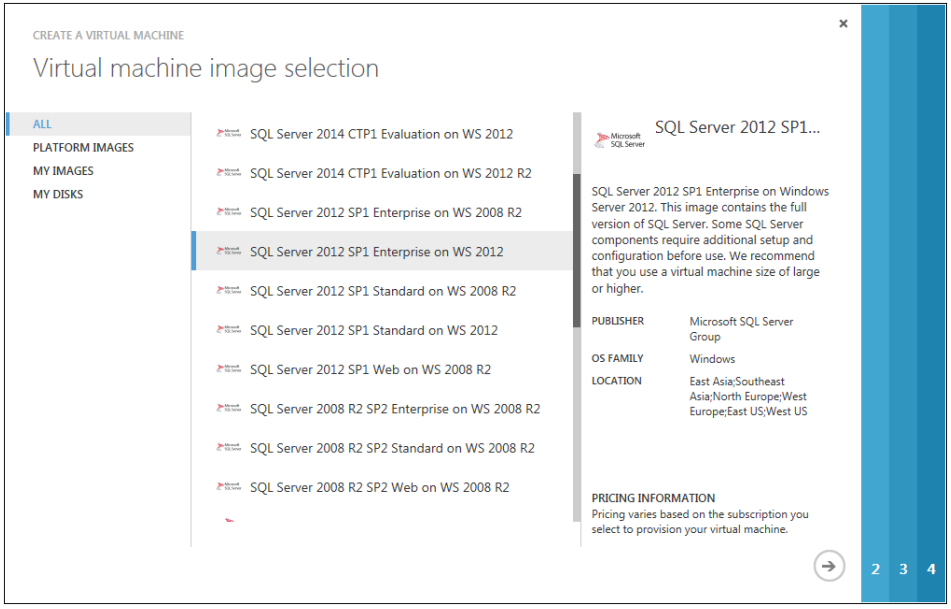


FIGURE 4-1 Windows Azure lets you create new virtual machines running different versions of SQL Server on different versions of Windows Server.

Hybrid solutions are easy to build as well using this approach. Combine on-premises SQL servers with ones running in the cloud to create multitier SQL Server applications for a wide range of solutions. Move your cloud-based SQL servers on-premises easily by migrating them from your datacenter into the cloud.

Windows Azure SQL Database

What if you want to realize the benefits of using SQL Server without the overhead of having to maintain and patch the operating system and applications? Windows Azure SQL Database can be an ideal solution in that case because it allows you to extend your business applications into the cloud by building on core SQL Server functionality while letting Windows Azure support staff handle the maintenance and patching tasks.

With Windows Azure SQL Database you can quickly create database solutions that are built on the SQL Server database engine. As Figure 4-2 shows, you can create a new SQL database in Windows Azure and then configure it later. You can decide whether to use an existing SQL database server or create a new one when you create your new database. You can also import a saved database from Binary Large Object (BLOB) storage into SQL Database.

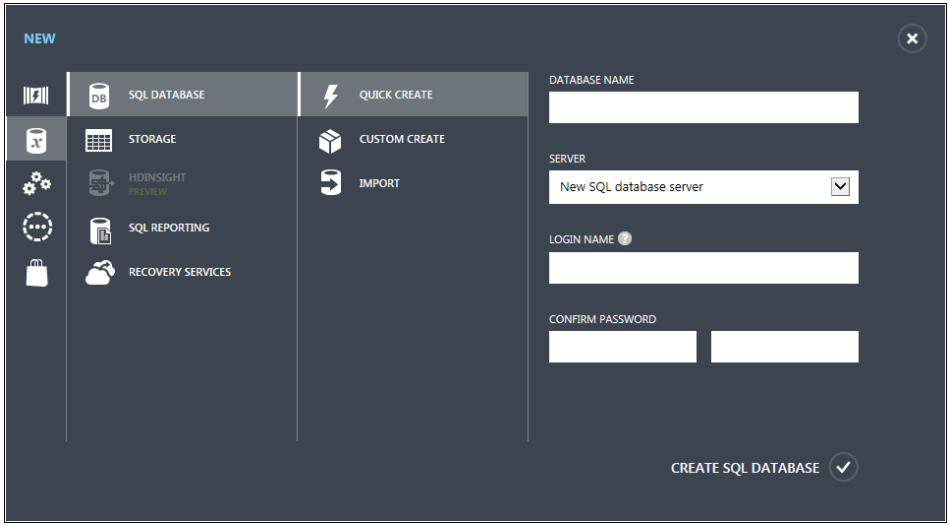


FIGURE 4-2 Windows Azure lets you quickly create a new SQL database in the cloud.

Once you've created your new database, you can use the Windows Azure Management Portal to log on to your database and create new tables, import data, create stored procedures, run queries, and perform similar database management tasks (see Figure 4-3).

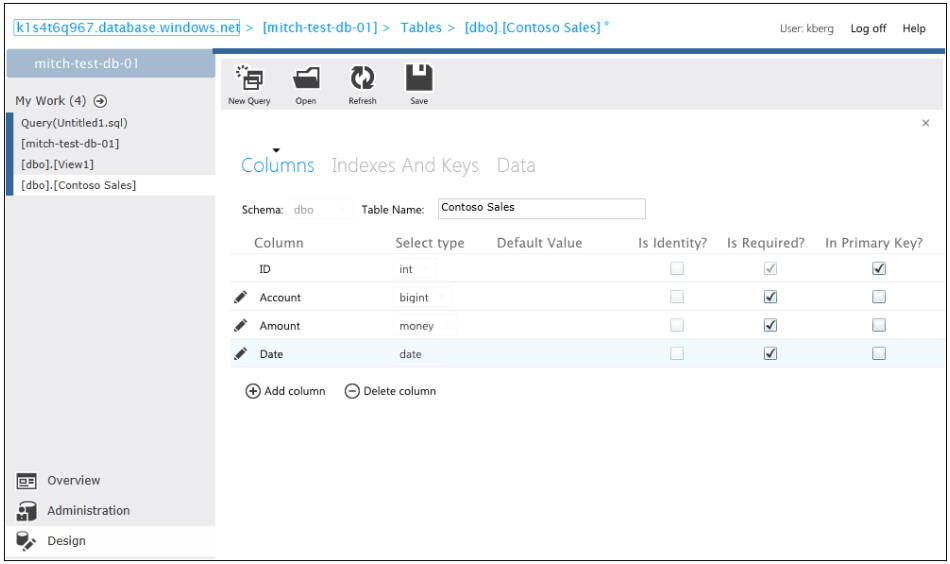


FIGURE 4-3 You can administer your SQL databases running in the cloud.

Your data is safe with Windows Azure SQL Database because it's stored in one primary datacenter and two replica datacenters. Your business can grow because SQL Database supports dynamically scaling out by federation database to up to 150 databases. And if you

need to guarantee the best level of performance for your application, you can reserve a fixed amount of database storage capacity at both the primary and secondary sites by using the new Premium version of the service.

IMPORTANT **It is important to understand that Windows Azure SQL Database is not feature-equivalent to Microsoft SQL Server, so not every exported SQL Server database can be imported successfully. Also, there are size limitations per database. For more information, see *http://msdn.microsoft.com/en-us/library/windowsazure/ff394115.aspx*.**

MORE INFO **The Premium version of Windows Azure SQL Database is currently in preview at the time of writing. For more information on this offering, see *http://www.windowsazure.com/en-us/pricing/details/sql-database/#service-premium*.**

Tables

NoSQL databases are nonrelational databases that don't support the standard T-SQL interface. Tables is a Windows Azure data management capability that can store large amounts of unstructured data. You can then access this data using REST application programming interfaces (APIs) either from within a service running in Windows Azure or directly over the Internet using HTTP/HTTPS request/response.

BLOB storage

BLOBs provide a simple mechanism for storing large amounts of text or binary data such as images, audio, or visual files. Windows Azure BLOB Storage can autoscale up to 200 terabytes and can be accessed using REST APIs in the same way as Tables. Applications running in Windows Azure can also mount a BLOB formatted as a single volume NTFS virtual machine which you can then move between private and public clouds using Windows Azure Drive.

To gain more insight into the capabilities of Windows Azure as an SQL Server database platform, let's now hear from one of our insiders at Microsoft.

Using Windows Azure Storage as a repository for SQL backups directly from within Microsoft SQL Server

Windows Azure represents a phenomenal shift in traditional enterprise thinking when it comes to how IT fundamentally delivers service to its internal business customers. While it sounds good on paper—many customers need to see a tactical example of the vision behind Azure in something that easily and rapidly transforms their thinking. Prior to joining Microsoft, I worked as a

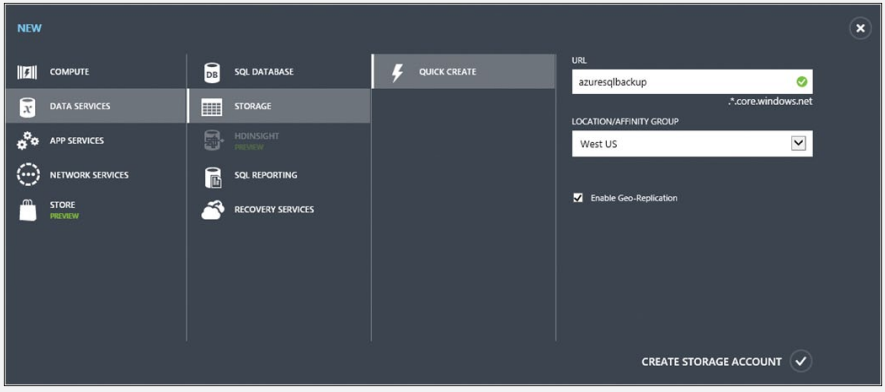
system administrator at various companies in the IT cost center providing jobs that traditionally consisted of delivering tried-and-true services to ensure the business continued running. Whether that was deploying servers, ensuring backups were running, or rotating out tapes, the fundamental job duties remained the same. I struggled to relate the benefit of what I did to how I affected the way that company did business—whether it was a university, pharmaceutical manufacturing company, or even a public utility. In addition, the cost benefit of the services IT provided were nebulous and hard to quantify.

With this experience and background, I enjoy talking to my customers about how Windows Azure can aid them in delivering rich services that used to be so hard to quantify and prove value—one of those being backups of SQL databases.

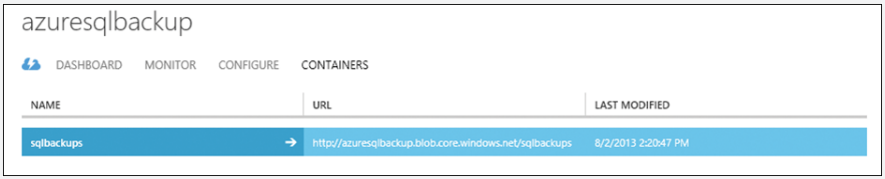
Traditionally, customers would purchase complex on-premises tape libraries, create schedules, install agents on SQL servers, and define backup windows that seemed to make sense. At some larger customers, SQL database administrators had no hand in the configuration of these backups and trusted their infrastructure system administrator colleagues to keep the service up and running. While there were good intentions all around, much of the control surrounding the integrity of this crucial IT process was spread too thin for there to be assurance that it was actually taking place, which resulted in outages and impacts on the business.

With SQL Server 2012 SP1 and Windows Azure, Microsoft has built into the core functionality of the product itself a method to leverage cloud scaling to change the way we think about backing up databases. No agents required. No access granted to SQL other than to the database administrators who know the data best. Native SQL backup commands allow backups to be stored directly in the cloud. The process is simple:

1. Create a storage account in Windows Azure. The storage account can be located in any Windows Azure datacenter across the globe.



2. Create a container in the storage account. The container is where you will store backups.



3. Copy the access key from the Windows Azure portal, and execute the backup command from within SQL, specifying the Windows Azure storage account and container.

```
-------Clear out Credential
DROP CREDENTIAL mycredential


-------Create Azure Storage Account Credential
IF NOT EXISTS
(SELECT * FROM sys.credentials
WHERE credential_identity = 'mycredential')
CREATE CREDENTIAL mycredential WITH IDENTITY = 'azuresqlbackup'
,SECRET = [redacted] ;


--------Perform Backup
BACKUP DATABASE [CriticalDatabase]
TO URL = 'https://azuresqlbackup.blob.core.windows.net/sqlbackups/CriticalDatabase-080213-Full.bak'
     WITH CREDENTIAL = 'mycredential'
    ,COMPRESSION
    ,STATS = 5;
GO
```

```
Messages
45 percent processed.
90 percent processed.
99 percent processed.
Processed 280 pages for database 'CriticalDatabase', file 'CriticalDatabase' on file 1.
100 percent processed.
Processed 2 pages for database 'CriticalDatabase', file 'CriticalDatabase_log' on file 1.
BACKUP DATABASE successfully processed 282 pages in 0.716 seconds (3.075 MB/sec).
```

With these three steps—and a total time to set up of about three minutes—I have just showed business value by doing the following:

- Created an offsite copy of a SQL database using native SQL functionality.
- Triple-replicated the backup of this database within the Windows Azure datacenter where the storage account lives.
- Geo-replicated the backup of this database to a datacenter at least 500 miles away. In this case, across the United States from the West coast to the East coast.

I purchased no equipment, installed no agents on the servers, did not have to train colleagues that are not SQL experts on the process, and can report to my management that the backup is offsite and guaranteed to be available with a 99.9 percent SLA. Furthermore, I pay only for what I use. If I no longer require this

backup, I can delete it instantly and no longer pay for utilization. I don't have to decommission equipment, burn tapes, or do anything else.

This functionality is a differentiator that shows Windows Azure is a core part of the Microsoft software DNA. Features like this, which augment the functionality of our mature software offerings, make me proud to be a Microsoft employee today and sharing this story with my customers to help them solve business problems.

You can find more information on MSDN at: *http://msdn.microsoft.com/en-us/library/jj919148.aspx*

Mike Gaal
Datacenter Technology Solutions Professional

Learn more

For general information about Windows Azure Data Management solutions and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/data-management/*.

For more detailed information on what Windows Azure SQL Database is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/sql-database/*.

For information on other Windows Azure Storage services, including Tables and BLOB storage, see *http://www.windowsazure.com/en-us/documentation/services/storage/*.

To get a deeper understanding of Windows Azure Tables, see *http://blogs.msdn.com/b/windowsazurestorage/archive/2010/11/06/how-to-get-most-out-of-windows-azure-tables.aspx*.

Finally, be sure to check out these TechEd 2013 presentations on Windows Azure SQL Database and Windows Azure Storage which are available for viewing and download from Channel 9:

- "Getting the Most Out of Windows Azure Storage," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B406*
- "Windows Azure Virtual Machines and SQL Unplugged," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B310*
- "Pushing Data to and from the Cloud with SQL Azure Data Sync," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B207*
- "Query Performance Tuning Techniques for Windows Azure SQL Database," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B337*

- "Windows Azure SQL Database for the DBA," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B323*
- "Cloud Optimizing T-SQL: What You Need to Know if You Use SQL Database or Microsoft SQL Server 2012," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B330*
- "Protecting Your Data in Windows Azure SQL Database," which is available at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B314*

HDInsight

Big data is a term used to describe business data stored in a combination of relational and nonrelational databases. The problem with big data is that it's difficult to analyze it when the data is stored in many different ways. How do you analyze data that is distributed across relational database management systems (RDBMS), XML flat-file databases, text-based log files, and binary format storage systems?

Hadoop, an Apache open source project, is the dominant technology used today for tackling the problem of analyzing big data. Hadoop lets you store data using the Hadoop Distributed File System (HDFS) and then analyze the data using MapReduce jobs spread across clustered servers.

The Windows Azure implementation of Hadoop is called HDInsight, and it fully supports the existing Hadoop ecosystem including Hive and Pig. HDInsight uses clustered virtual machines to store data and analyze data using HDFS and MapReduce jobs.

MORE INFO **HDInsight is currently in preview at the time of writing. For more information, see *http://www.windowsazure.com/en-us/documentation/services/hdinsight/?fb=en-us*.**

To learn more about HDInsight and how to use it, let's now hear from another of our insiders at Microsoft.

Monitoring solutions for HDInsight

Monitoring is an important part of system administration. In this section, we look at the monitoring facilities present in Windows Azure HDInsight and the Hadoop ecosystem, and how they can hook into external monitoring systems.

The purpose of monitoring is to detect when the cluster is not providing the expected level of service. The Namenodes and the Job Tracker are the most important to monitor, as their failure can be catastrophic. Failures on the task trackers and datanodes is to be expected, so you should provide extra capacity so that the cluster can tolerate having a small percentage of dead nodes at any time.

In addition to the tools described in this text, some administrators run test jobs on a periodic basis to test the cluster's health. Currently, this is widely used, as it allows administrators to collect fine-grained details on a cluster's performance. Moreover, the Hadoop ecosystem provides several tools to make this easier. The section "Using test jobs as monitoring tools" shows some of the tools provided to write such jobs.

There is a lot of work to add more monitoring capabilities to Hadoop, which is not covered here. For instance, Apache Chukwa is a Hadoop subproject devoted to large-scale log collection analysis. Logging techniques are also not presented here. All Hadoop daemons produce log files that are very useful for finding out what is happening in the system. For more information refer to page 256 of the book *Hadoop: The Definitive Guide* from O'Reilly Media.

Built-in monitoring solutions

The HDInsight platform provides an easy-to-use cluster monitoring tool located on the main dashboard under Monitor Cluster. The monitoring tool shows the status of the Namenode and Job Tracker as well as the health of every worker in the cluster. Moreover the built-in tool provides historic map-reduce metrics (such as the number of succeeded and failed map and reduce tasks), job submission data (including number of jobs started and failed,) and queue information (such as number of map and reduce waiting tasks).

HDInsight also provides a job history of jobs submitted through the dashboard (note that it does not include jobs submitted using the Hadoop command line or the JavaScript console). The Job History page shows the job name, the start and end time, the Hadoop command line request, and the status of the job. Clicking the desired job will show the job type, exit code, and output. Historical job information is located on the main dashboard under Job History.

Developers can also find information on jobs through the WebHCat API available on HDInsight. However, we will not cover the API here, as it only offers limited information on jobs. You can find more about it at: *http://www.windowsazure.com/en-us/manage/services/hdinsight/howto-execute-jobs-programmatically/*

Apache Ambari

The Ambari project is aimed at making Hadoop management simpler by providing a framework for provisioning, managing, and monitoring Hadoop clusters. Ambari provides a dashboard for monitoring the health and status of a cluster. Moreover, it leverages Ganglia, a distributed monitoring system for clusters, and Nagios, an alerting system for clusters, to provide real-time cluster metrics that are specific to a Hadoop environment. Ganglia is good for collecting several metrics and graphing them, while Nagios is good at sending alerts when a critical threshold is reached in any set of metrics. Both tools combine to make Ambari an excellent Hadoop monitoring framework.

At the time of this writing, Ambari is not available for HDInsight. However, Microsoft and Hortonworks, a provider of enterprise Hadoop solutions, are currently collaborating on a Windows-specific implementation of Ambari. Also, Hortonworks will be releasing a management pack for Microsoft System Center Operations Manager and Microsoft System Center Virtual Machine Manager (VMM) that will leverage the Ambari framework and integrate its functionalities with newer versions of System Center.

Using test jobs as monitoring tools

Many times the necessary monitoring information is not available through built-in or external tools. In such cases you must build your own tool to extract the necessary data from the cluster. Note that there are many ways of collecting custom data from a cluster. For instance, the log files issued by the daemons can be searched, the cluster can be reconfigured to expose certain parts (such as the Job Tracker), etc. This section aims to walk you through the one suitable way of collecting data in newly deployed HDInsight clusters.

The process of collecting custom data from the Hadoop framework in HDInsight is fairly simple and is outlined below:

1. The administrator (or an external tool) submits a Map Reduce job to the cluster.
2. The cluster gets a hold of the Job Tracker via the Job Client.
3. The Job Client returns job and task related information.
4. The Map Reduce job uploads the acquired data to a BLOB.

Now we will walk you through the creation of a tool that will collect all present information for every job and upload it to a Windows Azure container. The code presented has exception handling and imports omitted to save space. The steps to write the job are shown in Java, as that is the language for the Hadoop framework. However, you may write your Map Reduce jobs in C# using the .NET Hadoop SDK (*http://hadoopsdk.codeplex.com/*). As we only need information from the Job Tracker, we won't need to run our tool on every node of the cluster. We just need to have access to the Job Tracker. On HDInsight, the Master node has access to the Job Tracker, so we will just need to run our tool at the master. To make it easier, we can make our tool into a Map Reduce job that only runs on the master. That way we can submit our tool externally through the dashboard and have our BLOBs updated.

Usually a Map Reduce job will contain three components: a Map class, a Reduce class, and a main function. Since we only need to run on the Master node, we won't need either the Map or Reduce classes. The entry point to our application will look like this:

```
public class MainClass {
```

```
    public static void main(String[] args) {
       // Job tracker connect info
          String jobTrackerHost = "YourJobtrackerHost";
          int jobTrackerPort = "YourJobtrackerPort";

        // Storage connect info
        String storeAccountName   = "YourStoreAccountName";
        String storeAccountKey    = "YourStoreAccountKey";
        String storeContainerName = "YourContainerName";

        // Define both a gatherer and an uploader
        // we'll create these classes later
        JobGatherer   gatherer = new JobGatherer(jobTrackerHost,
jobTrackerPort);
        AzureUploader uploader = new AzureUploader(storeAccountName,
storeAccountKey, storeContainerName);

        // For every JSON representation of a gathered job
        // upload a new blob with the Job ID as a name and
        // the job information a value
        for (JSONObject job : gatherer.getAllJobs()) {
          uploader.uploadString(job.getString("JobId"), job.toString());
        }
    }
}
```

Our gathering class will have two main functions: acquire the necessary jobs and serialize them to JSON. We will make use of Hadoop's libraries to create a JobClient object that will connect to the Jobtracker. Our JobClient will then acquire all jobs submitted and return them in JSON format to our main class. Make sure you add Hadoop library dependencies to your project for it to work!

```
public class JobGatherer {

  private JobClient jobClient = null;

  public JobGatherer(String jobTrackerHost, int jobTrackerPort) {
          // Connect to the jobtracker
          jobClient = new JobClient(new InetSocketAddress(jobTrackerHost,
jobTrackerPort), new Configuration());
  }

  // Get all submitted jobs and serialize them
  public Vector<JSONObject> getAllJobs() {
```

```
        Vector<JSONObject> vect = new Vector<JSONObject>();
        // Object writer for serialization
        // Present in the Jackson JSON library for Java
        // You can find more about it at: http://jackson.codehaus.org
        ObjectWriter ow = new ObjectMapper().writer().
withDefaultPrettyPrinter();

        for (JobStatus job : jobClient.getAllJobs()) {
                vect.add(new JSONObject(ow.writeValueAsString(job)));
        }
        return vect;
  }
```

Finally, we just need to write our uploader. The uploader's job is to connect to your Windows Azure storage account and upload the JSON data as BLOBs. Microsoft provides a Windows Azure SDK for Java, which we will be using. You can find more about it at: *http://www.windowsazure.com/en-us/develop/java/how-to-guides/blob-storage/*.

```
public class AzureUploader {
  // Default settings
  private static final String EndopointProtocol = "DefaultEndpointsProtoc
ol=http";
  private static final String DEFAULT_ENCODING  = "UTF-8";

  // Client's data
  private static String AccountName = null;
  private static   String AccountKey = null;
  private static String StorageConnection = null;

  // Azure specific connections
  private static CloudStorageAccount storageAccount = null;
  private static CloudBlobClient blobClient = null;
  private static CloudBlobContainer container = null;


  public AzureUploader (String accountName, String accountKey, String
containerName) {
        AccountName = accountName;
        AccountKey = accountKey;
        StorageConnection = EndopointProtocol + ";AccountName=" +
AccountName +
                        ";AccountKey=" + AccountKey;

        storageAccount = CloudStorageAccount.parse(StorageConnection);
```

```
            blobClient = storageAccount.createCloudBlobClient();
            container = blobClient.getContainerReference(containerName);
            container.createIfNotExist();
    }

    public void uploadString(String name, String data) {
            CloudBlockBlob blob = container.getBlockBlobReference(name);
            InputStream in = new ByteArrayInputStream(data.
getBytes(DEFAULT_ENCODING));
            blob.upload(in, data.length());
    }
}
```

Further information can be extracted from jobs. Refer to the Hadoop documentation on JobStatus (*https://hadoop.apache.org/docs/stable/api/org/apache/hadoop/mapred/JobStatus.html*) for available information.

Monitoring workflows: The Oozie API

Many times a single Hadoop job will not be enough to complete the task necessary. That's why Hadoop has the concept of workflows. Workflows are Directed Acyclic Graphs of actions and dependencies. Actions can be Map Reduce jobs, Pig applications, Hive queries, etc. Dependencies indicate the order that actions are executed. For more about workflows, see *http://archive.cloudera.com/cdh/3/oozie-1.6.2+57/WorkflowFunctionalSpec.html*.

Often you'll need to monitor the performance of given workflow. For instance, with two equivalent workflows, it would be nice to determine which one runs faster on your current cluster configuration. Thankfully, Oozie, Hadoop's workflow scheduler system, provides an HTTP REST API to manage and submit jobs. The Oozie API allows developers to find which actions were executed on a given workflow, the time taken to run that workflow, and much more. For more about the Oozie API visit: *http://oozie.apache.org/docs/3.1.3-incubating/WebServicesAPI.html*.

Paulo Almeida Tanaka
SDET Intern at Microsoft Mediaroom

Learn more

For general information about HDInsight and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/hdinsight/*.

For more detailed information on what HDInsight is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/hdinsight/*.

Finally, be sure to check out these TechEd 2013 presentations that cover HDInsight which are available for viewing and download from Channel 9:

- "HDInsight: Introduction to Hadoop on Windows," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B221*
- "Data Management in Microsoft HDInsight: How to Move and Store Your Data," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B334*
- "Predictive Analytics with Microsoft Big Data," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B339*
- "Big Data Analytics with Microsoft Excel 2013," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/DBI-B336*

Business Analytics

Besides Hadoop, Windows Azure also includes other business analytics solutions you can use for discovery and data enrichment. For example, Windows Azure Virtual Machines allows you to deploy and use SQL Server business analytics technologies like Reporting and Analysis Services and Microsoft SharePoint Server in virtual machines running in the cloud. Organizations that need to maintain control over their business analytics solutions can build hybrid solutions with data and applications that span both on-premises databases and storage in the Windows Azure public cloud. But what if you don't want the hassle of maintaining and patching the virtual machines providing your business analytics solution?

Windows Azure SQL Reporting

Windows Azure SQL Reporting is a cloud-based service that lets you use easily add reporting capabilities into your Windows Azure application. As Figure 4-4 shows, you can create a new SQL Reporting service you can use to create reports with tables, charts, maps, and other reporting capabilities.

Once you've created your reporting service you can upload and download reports, run reports from the Management Portal, create new data sources and folders, and perform other SQL Reporting administration tasks.

Windows Azure Marketplace

The Windows Azure Marketplace, shown in Figure 4-5, is an online market where you can buy and sell finished Software as a Service (SaaS) applications and premium data. Using the Windows Azure Marketplace you can commerce-enable your applications and achieve global reach by being able to complete transactions in multiple geographies and currencies.

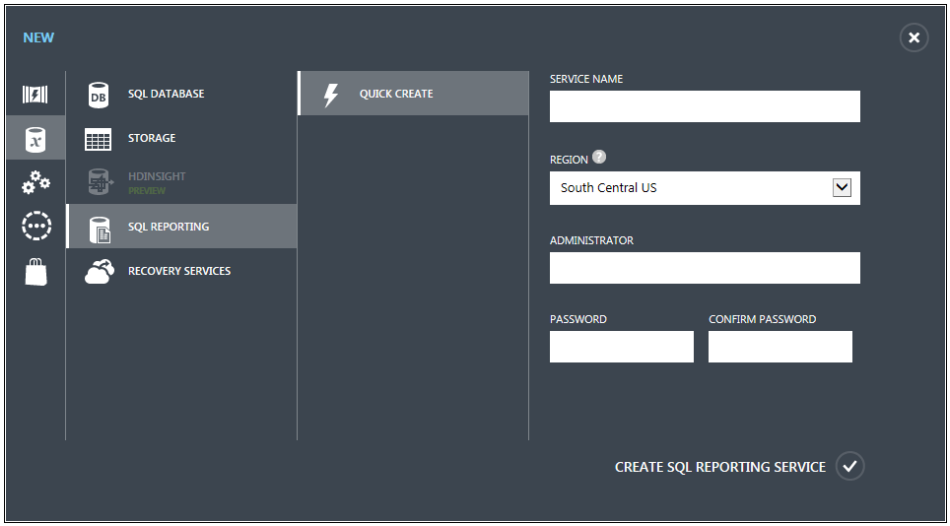


FIGURE 4-4 You can use Quick Create to make a new SQL Reporting service in Windows Azure.

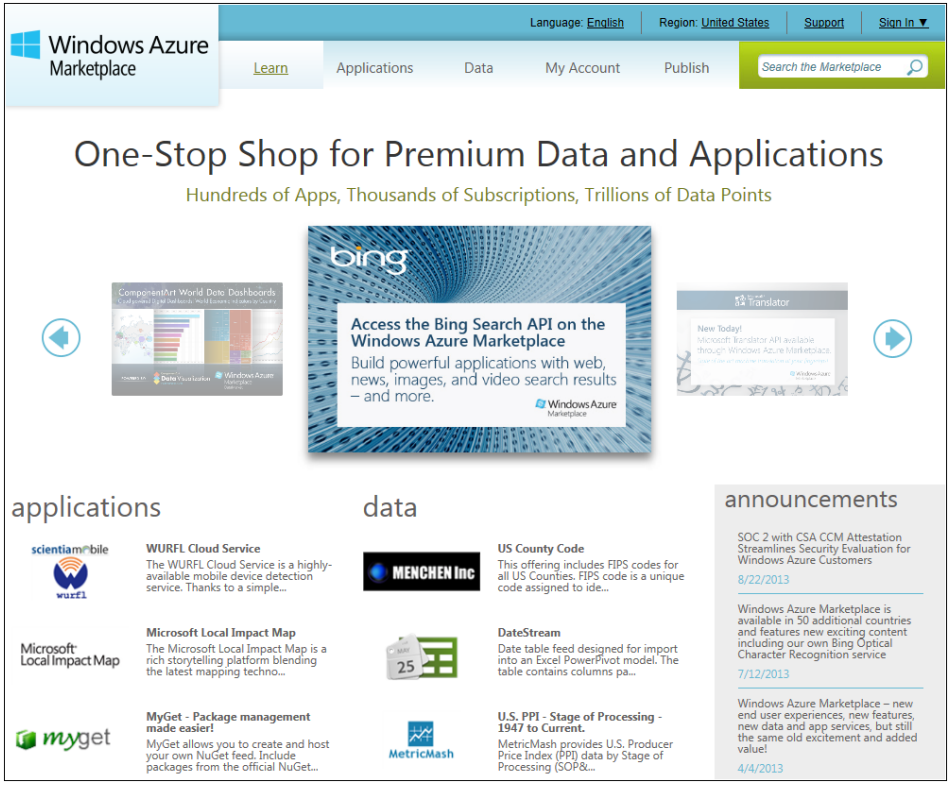


FIGURE 4-5 The Windows Azure Marketplace.

For data, you'll find a wide variety of data on the Windows Azure Marketplace including demographics, environment, financial, retail, and sports. You can use this data in your Microsoft Office software, business intelligence (BI) tools, and your very own custom applications. And for applications, you can use the Windows Azure Marketplace to discover, try, and buy applications built on the Windows Azure platform and purchased through a single trusted source.

MORE INFO **Experience the Windows Azure Marketplace today by visiting *https://datamarket.azure.com*.**

Learn more

For general information about Windows Azure Business Analytics and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/business-analytics/*.

For detailed information on what Windows Azure SQL Reporting is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/sql-reporting/*.

To learn more about the Windows Azure Marketplace, see *https://datamarket.azure.com/about*.

Backup

Backups are of primary importance when it comes to protecting your valuable business data. Automating the backup process is essential to ensure you always have your data backed up in the event of a disaster. Storing backups offsite is another best practice because it eliminates a single point of failure for data restoration.

Windows Azure Backup (currently in preview at the time of writing) together with the Windows Server and System Center platforms let you automate the backup of your business data to the cloud. For example, you can configure Windows Azure Backup to back up files and data from your on-premises Windows Server systems to the Windows Azure cloud. You can use the Windows Azure Backup Agent to specify a backup schedule for your registered servers and to recover files and folders from the cloud if a problem prevents you from accessing your physical servers.

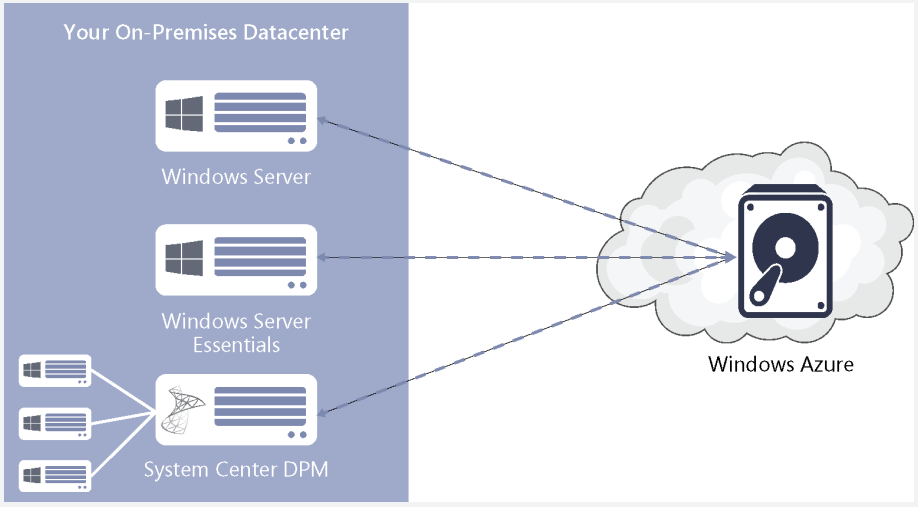
You can also use System Center Data Protection Manager (DPM) together with Windows Azure Backup to protect your data. For example, instead of backing up your primary DPM servers to disk or to secondary on-premises DPM servers, beginning with System Center 2012 SP1 DPM you can now back up your DPM servers and the data protected by those servers to the cloud using Windows Azure Backup.

MORE INFO **Windows Azure Backup is currently in preview at the time of writing. For more information, see *http://www.windowsazure.com/en-us/services/backup/*.**

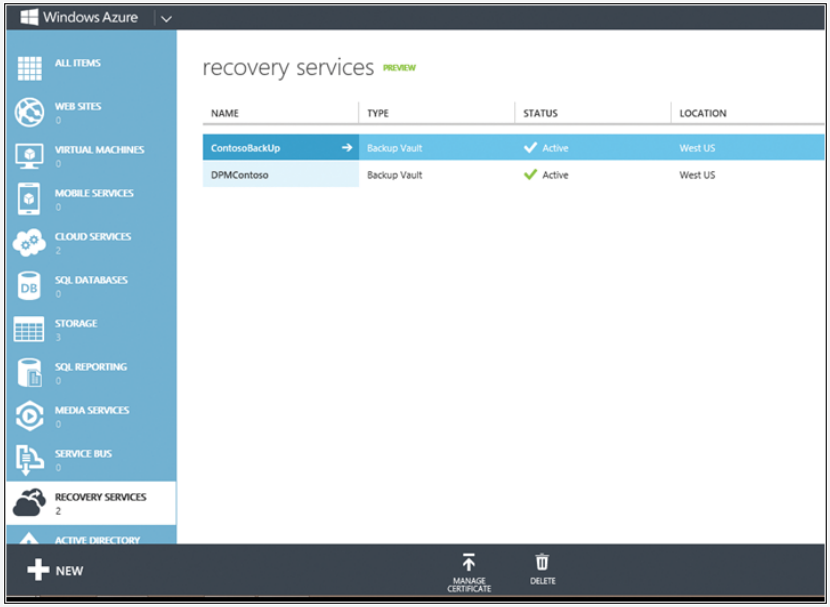
Let's follow along now as one of our Microsoft insiders walks us through the steps of how Windows Azure Backup works.

Windows Azure Backup leverages the cloud to protect your data

Customers have being using onsite backup solutions to protect their Windows Servers since the initial release of Windows NT 3.1. Today there are many different types of technologies onsite and offsite to protect your data. Microsoft has released an add-on product called Windows Azure Backup—as the name suggests, it backs up your data to the cloud! The great feature of Windows Azure Backup is that you can use the same technology to protect data in a virtual machine running on-premises, in a private or public cloud, as well as a physical server. All you need is Windows Server 2008 R2 SP1 or higher as your operating system and a Windows Azure account with the Windows Azure Backup feature enabled. Windows Azure Backup can even back up your System Center 2012 SP1 DPM data to the cloud!



Windows Azure Backup works by installing a Windows Azure Backup Agent on the Windows Server, Windows Server Essentials, or System Center 2012 Sp1 DPM server. You can also use Windows PowerShell cmdlets to protect data. This agent backs up data to a backup vault that is configured in Windows Azure, under Recovery Services.



Requirements, configuration, and management of protected content

Recently, a customer's proof of concept proved the simplicity of the requirements and configuration. You need a Windows Azure subscription with a backup vault configured under Recovery Services, a Windows Server or System Center 2012 DPM server to protect, a certificate, and a network connection.

For the proof of concept mentioned earlier, I used my MSDN Windows Azure benefits. Then I configured a backup vault through the portal. You just choose Recovery Services |Create New Recovery Services | Backup Vault |Quick Create and then enter a name and a location. You now have a backup vault that is created in Windows Azure storage for your online backups.

You have to use a certificate to create a secure connection between your server and Windows Azure Backup. This certificate has to be either a self-signed certificate or any valid SSL certificate issued by a Certification Authority (CA) trusted by Microsoft, whose root certificates are distributed via the Microsoft Root Certificate Program. In this case I created a self-signed certificate for my server RD01.Contoso.com that I wanted to back up. Here is the command:

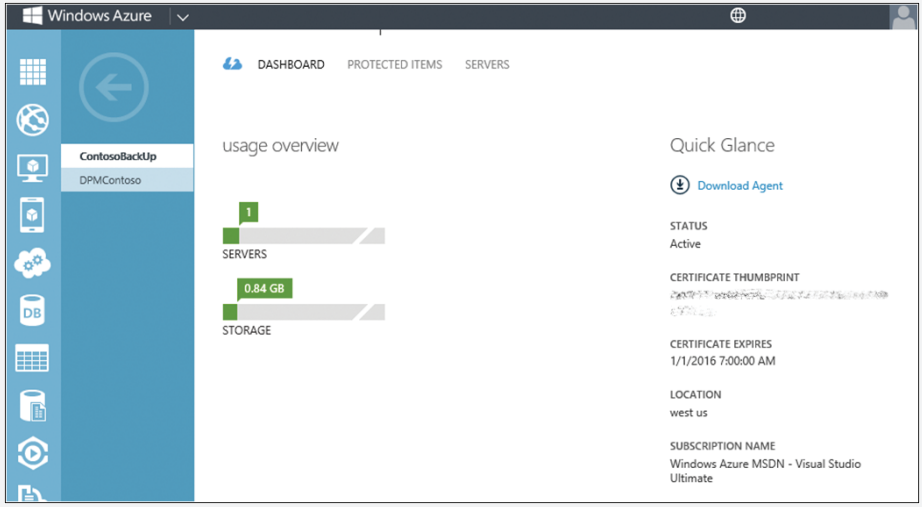
```
C:\Azure> makecert.exe -r -pe -n CN=RD01.Contoso.com -ss my -sr
localmachine -eku 1.3.6.1.5.5.7.3.2 -len 2048 -e 01/01/2016 RD01.cer
Succeeded

C:\Azure>
```

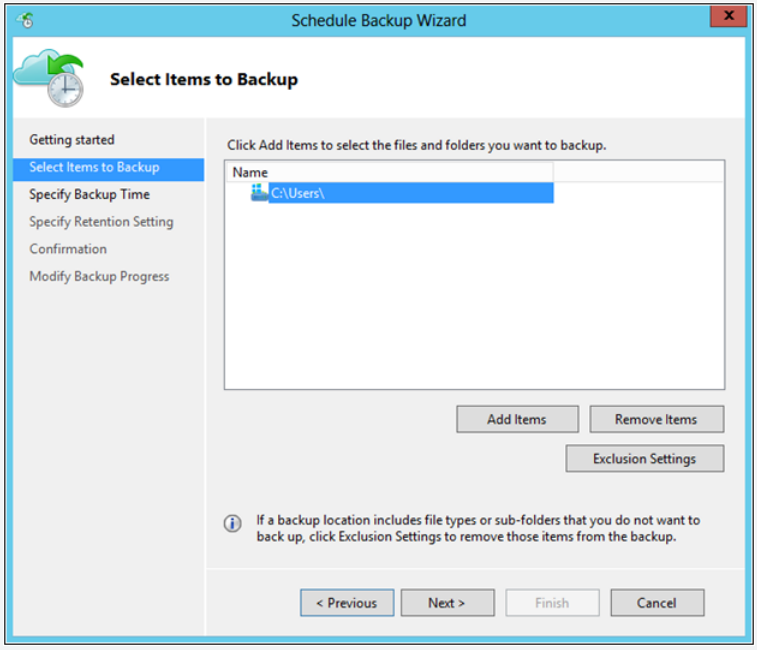
NOTE **The certificate has to have a valid ClientAuthentication EKU, and it has to be currently valid with a validity period that does not exceed 3 years.**

You then upload that certificate to your backup vault by choosing Manage Certificates on the backup vault and selecting the certificate you created or your issued certificate.

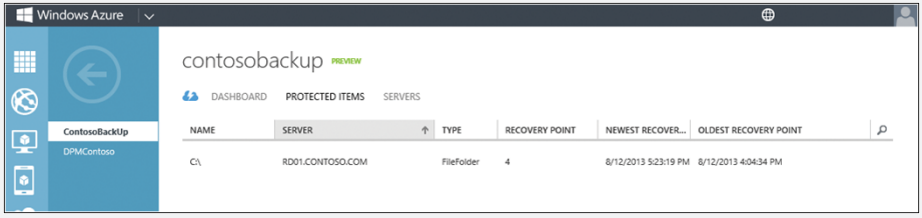
Once you have your certificate uploaded, you then need to install the Windows Azure Backup Agent in your server. Installing the Windows Azure Backup Agent is very simple in itself. You can download it directly from your Windows Azure backup vault Dashboard.



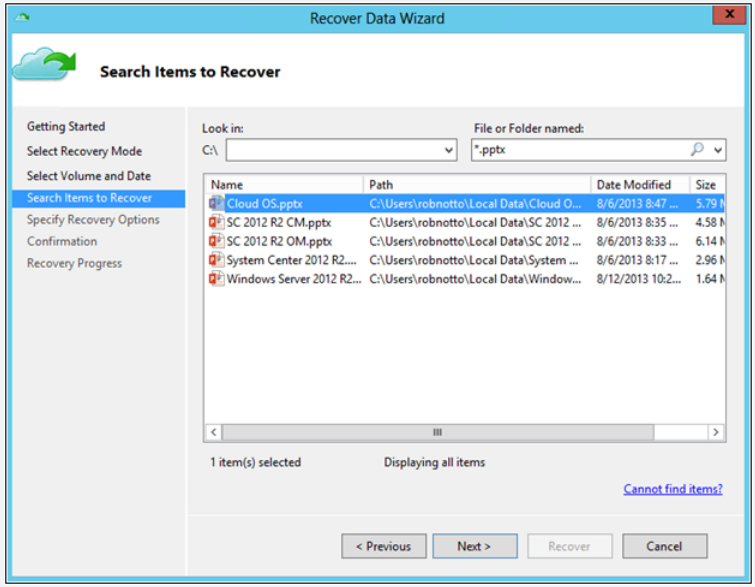
Now that you have the Windows Azure Backup Agent installed you must register the server through the Windows Azure Backup console or through Windows PowerShell. The Windows Azure Backup console is installed as part of the Windows Azure Backup Agent installation, as are the Windows PowerShell cmdlets. To back up your content, you choose the data you want protected to the cloud. This is also done through the Windows Azure Backup console or Windows PowerShell. Because we use incremental backups, only changes to files are transferred to the cloud. This helps ensure efficient use of storage, reduced bandwidth consumption, and point-in-time recovery of multiple versions of the data. Configurable data retention policies, data compression, and data transfer throttling also offer you added flexibility and help boost efficiency. Selecting the content, backup type, and retention period is completed very easily through the Schedule Backup Wizard.



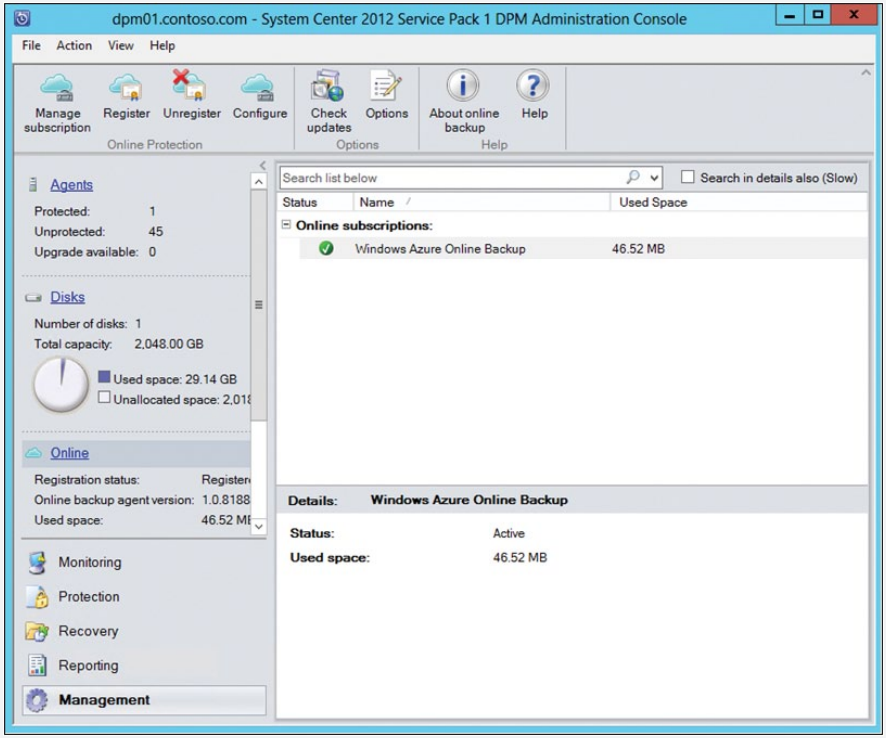
The Dashboard view in the Windows Azure Management Portal provides a list of protected items and possible recovery points. The Dashboard will also give you views of how much data you're protecting and the servers that are being protected.



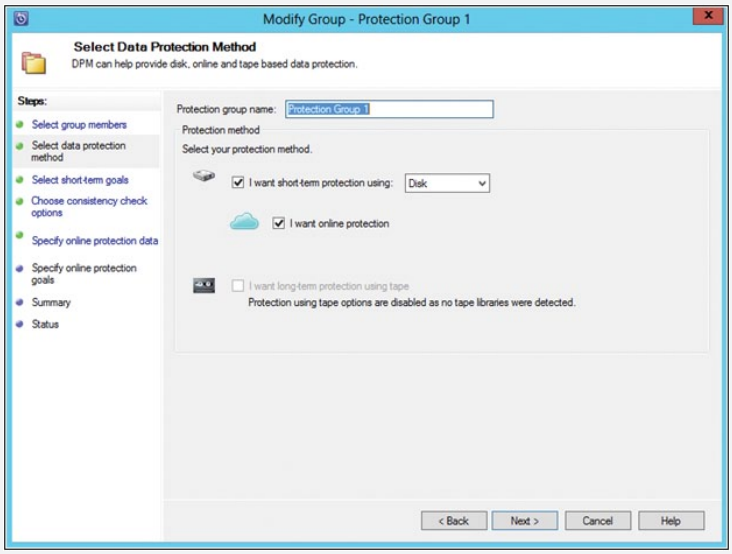
Recovery of content is easy and straight forward also. Just like backing up data, the recovery of data is completed through the Windows Azure Backup console or Windows PowerShell cmdlets.



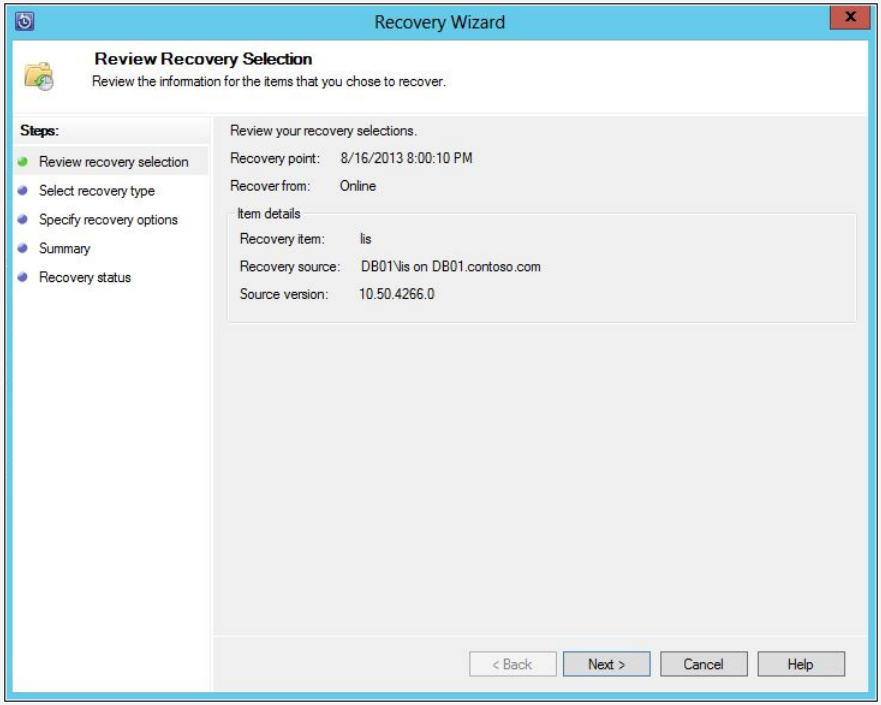
It takes under 30 minutes to get this solution fully operational. I also have set up System Center 2012 SP1 DPM for online protection via Windows Azure Backup. This was as easy as installing the Windows Azure Backup agent in my DPM server, and then registering the server with the backup vault. The requirements and configuration are almost exactly the same as configuring a server for Windows Azure Backup with the primary difference being that you have to register the System Center 2012 DPM server through the DPM Administration console.



To set up online protection, you choose it as an option when you run the setup protection via the Protection wizard.



Recovery works the same way—you can choose from an online or disk-based recovery point.



When I talk to customers about Windows Azure Backup, their concerns are costs, security, and the types of data they should protect with this solution?

I talk to customers every day about Microsoft solutions for managing private, public, and hybrid clouds. The most common concerns are costs, security, and the types of solutions available for use in public clouds.

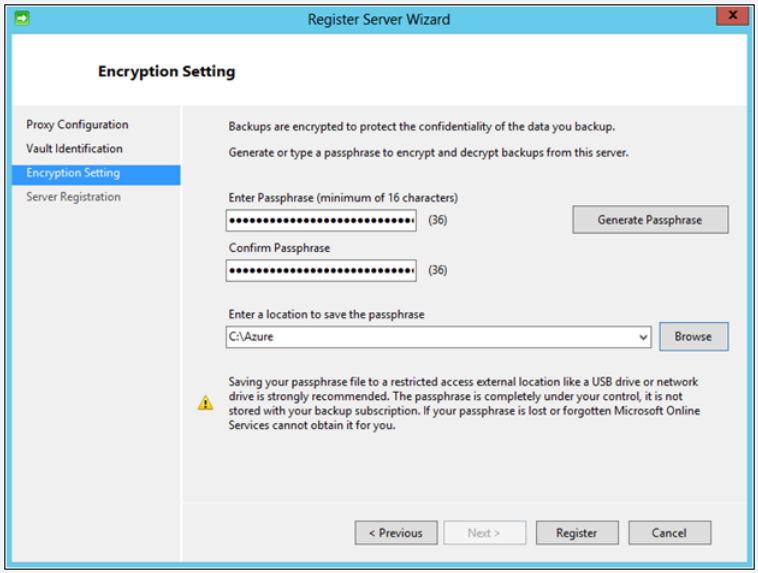
Microsoft charges for backup based on the amount of data stored in the Backup service. Microsoft does not charge additionally for bandwidth, storage, storage transactions, compute, or other resources associated with providing the Backup service.

The following price example is based on a 50 percent preview discount.

COMPRESSED DATA STORED PER MONTH	PRICE (PREVIEW)
First 5 GB / Month	Free
Greater than 5 GB / Month	$0.25 per GB per month

The charge for backup is prorated daily. For example, if you consistently utilized 20GB of storage for the first half of the month and none for the second half of the month, your daily average amount of compressed data would be 10Gb for the month. The first 5GB is free so your bill under the preview pricing would be $1.25 (5GBX$.025) for that month! The amount of storage you are billed is determined by the compression ratio and the number of backups retained.

Security is another concern when storing your data online. Windows Azure Backup uses certificates to create a secure connection between the server and the Windows Azure backup vault. In addition, all the data is encrypted before it is sent to Windows Azure. In order to do this, Microsoft uses a passphrase that you enter (or have generated) during the server registration process. The data is stored in Windows Azure Backup in an encrypted state. You can see the warning included in the Wizard in the following graphic!



Many customers also ask about the types of data you can protect with Windows Azure Backup. Files and folders can be protected. You cannot back up system state!

Using System Center 2012 SP1 DPM you can back up files, Hyper-V virtual machines, and SQL Server databases.

Organizations should consider Windows Azure Backup if they are running virtual machines in Windows Azure, using System Center 2012 SP1 DPM, or have a need to store Windows Server data offsite for recovery purposes.

Robert Nottoli
Technology Solution Prof., US - MidWest - STU Core Infra

Learn more

For general information about Windows Azure Backup and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/backup/*.

For more detailed information on what Windows Azure Backup is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/recovery-services/*.

For a tutorial on how to get started using Windows Azure Mobile Services, see *http://www.windowsazure.com/en-us/develop/mobile/tutorials/get-started/*.

Also be sure to see the TechEd 2013 presentation titled "Automate Private Cloud Protection and Recovery with Microsoft System Center 2012 - Data Protection Manager," which is available for viewing and download from Channel 9 at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B401*.

Recovery Manager

If you've deployed a private cloud solution using System Center 2012, you can use Windows Azure Hyper-V Recovery Manager to protect the sensitive business data stored in your private cloud. Hyper-V Recovery Manager works by coordinating the replication of the virtual machines that comprise your private cloud to a secondary location. Replication is ongoing and asynchronous in nature, and you can monitor its progress using Recovery Manager.

In the event of a disaster causing an outage at your primary site, your virtual machines can be recovered and brought online in an orchestrated fashion to help you get your private cloud up and running again quickly. For audit and compliance reasons, you can use this service to test the recovery process and to temporarily transfer cloud services between your primary and secondary sites.

MORE INFO **Hyper-V Recovery Manager is currently in preview at the time of writing and participation in the program is limited. To apply for acceptance in the program, visit *http://connect.microsoft.com/datacenterrecovery/Survey/Survey.aspx?SurveyID=15188&clcid=0x409*.**

We'll conclude this chapter with a walkthrough of Hyper-V Recovery Manager by one of our experts inside Microsoft.

Hyper-V Recovery Manager leveraging the cloud to protect your important services

I talk to organizations daily about the benefits of cloud computing. Many customers have been evolving their virtualized datacenters to private cloud implementations. Private cloud implementations promise to bring enterprises the next round of datacenter optimizations like those that were gained going from physical servers to virtualized servers. As organizations evolve to private clouds there is a need to evolve their disaster recovery solutions also. Customers have repeatedly stated that current cloud-based disaster recovery solutions are complicated to set up, maintain, and validate in addition to being very costly. Microsoft, in designing Hyper-V Recovery Manager, has made it a point to deliver a solution that is easy to set up, maintain, and validate for protecting Hyper-V cloud implementations.

Hyper-V Recovery Manager is a Windows Azure service that manages cross-site protection and recovery of System Center VMM-based clouds. The key benefits Hyper-V Recovery Manager directly provides also addresses the concerns and challenges that organizations face today in building cloud-based protection.

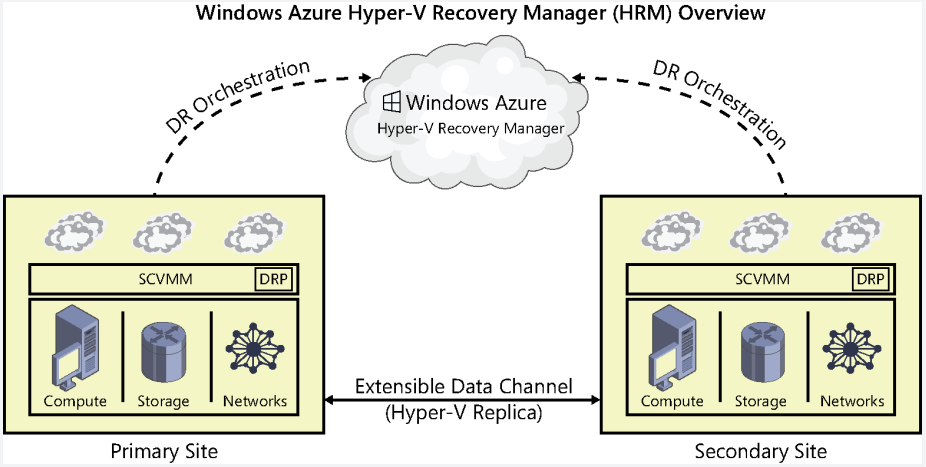
Some of the key benefits of Hyper-V Recovery Manager include:

- **Simplified configuration of virtual machine protection across datacenters**
- **Reliable cloud-based recovery plans for application failover**
- **Consistent use experience for remote management**
- **Extensible from the ground up**

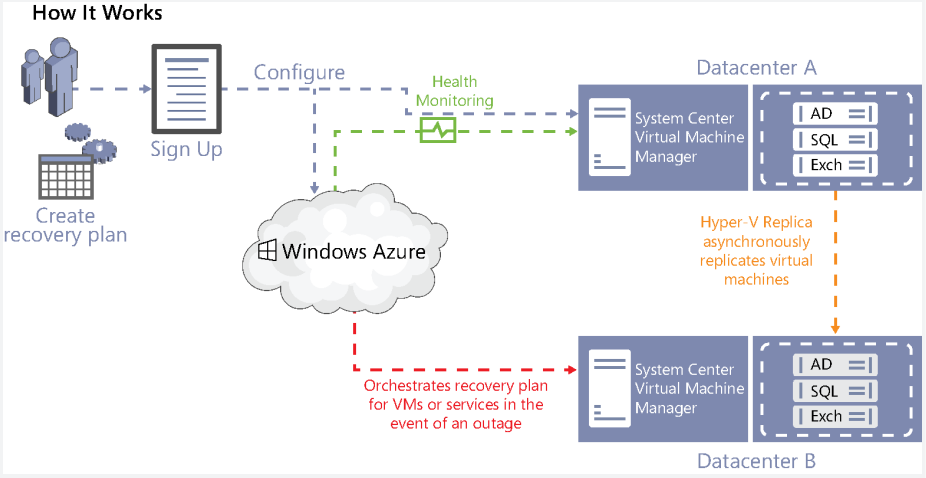
Most customers are interested at the point when I say "simple at scale" and want to know more. How does Hyper-V Recovery Manager accomplish this?

- **Windows Azure Hyper-V Recovery Manager protects important services by coordinating the replication of virtual machines at a secondary location.**
- **Windows Azure Hyper-V Recovery Manager extends site-to-site protection to entire private clouds by leveraging the Hyper-V Replica asynchronous virtual machine replication capabilities in Microsoft Windows Server 2012. It monitors the state of these clouds to ensure they are running.**

- In the event of a failure, Windows Azure Hyper-V Recovery Manager orchestrates the recovery of virtual machines at your secondary location. It does this by allowing you to build recovery plans and store them in Windows Azure.



At this point when I am talking to customers about Hyper-V Recovery Manager, most of them want to know how Hyper-V Recovery Manager works. The following diagram walks you through the moving parts.



To set up the service you need to download a single provider and run a wizard on the System Center 2012 VMM servers. This provider runs in the context of System Center 2012 VMM and communicates with the Hyper-V Recovery Manager service through an encrypted channel. No additional software or complex configurations are required.

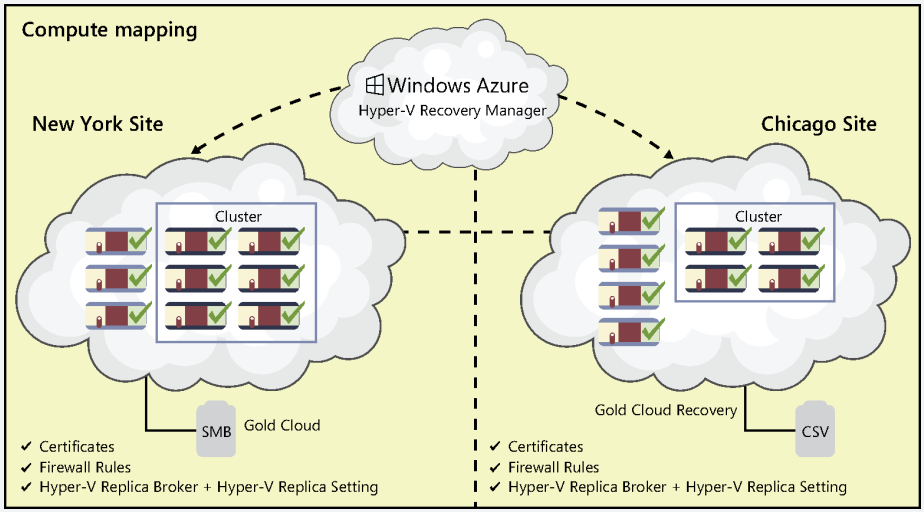
Instead of installing and configuring a complex software implementation, this service is already built into Windows Azure.

You configure the Hyper-V Recovery Manager via a console in Windows Azure. Again, the key point is that you do not need to install and configure a complex software implementation onsite. It is already a part of Windows Azure.

You configure "recovery plans"—the orchestration plan on what gets failed over and how.

Let's use this example. You have a New York site and a Chicago site. In the New York site you have a cloud called "Gold." In Chicago, you create a cloud called "Gold Cloud Recovery." You then choose that you want to failover the Gold cloud in New York to Gold Cloud Recovery in Chicago. This mapping of what you protect is referred to as protected items. Now that you have done that, Hyper-V Recovery Manager does all the heavy lifting. It configures all the hosts that are part of both the Gold and Gold Cloud Recovery clouds with the required certificates and firewall rules, and it configures the Hyper-V Replica settings.

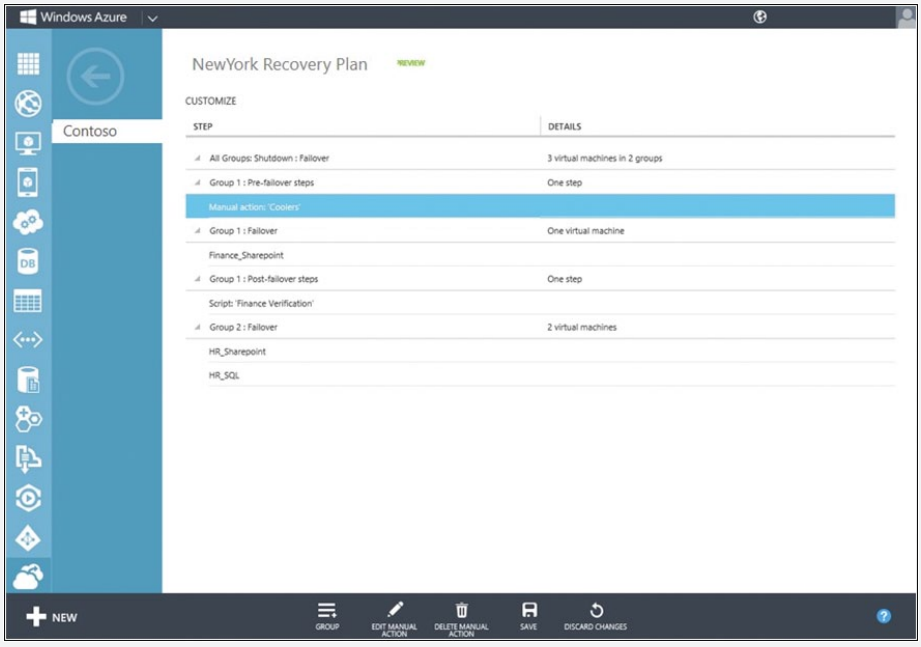
So, the end result is all the virtual machines running in the New York Gold cloud are configured and ready to be failed over to the Chicago Gold Cloud Recovery.



After that you map the networks of New York site to networks of Chicago site so that when the virtual machines failover, the service will connect them to the right networks, ensuring business continuity.

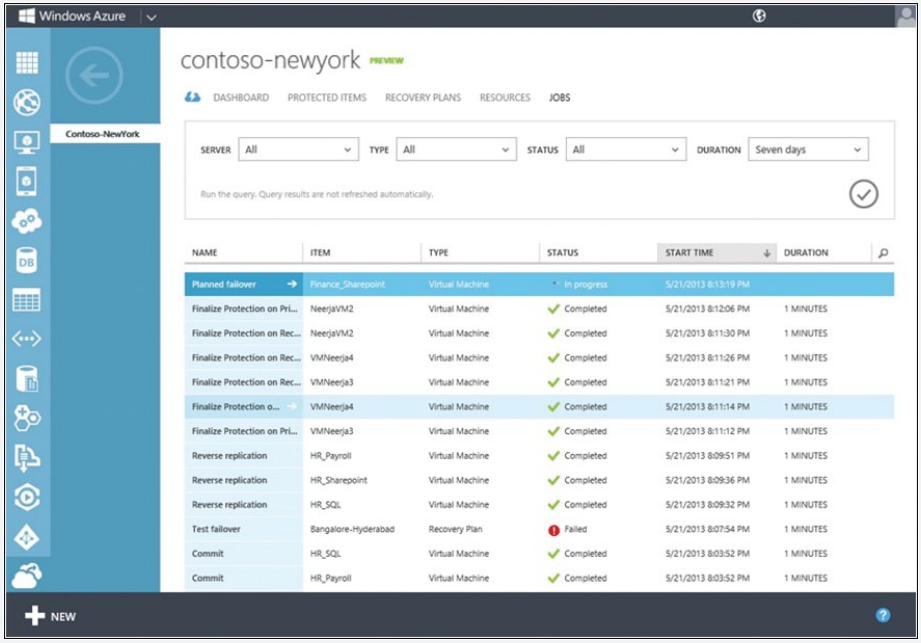
Next, you create recovery plans that detail what you want to have happen in the case of a Disaster Recovery. Below you will find a recovery plan for New York. You can see that all groups failover over to Chicago when they experience a shutdown as follows:

- Group 1, which has a SharePoint virtual machine for finance has a pre- and post-failover step.
- Group 2, which has a SharePoint and SQL virtual machine for HR, performs a straight failover.



You can also see from the recovery plans that you can create groups. Groups are a collection of virtual machines that failover together and groups themselves failover in sequence. For example, virtual machines in Group1 fail over first together, followed by virtual machines in Group 2. For each group, you can have pre or post steps that can be automatic or manual for performing actions like changing the DNS settings of the service to reflect a new IP address of the service. Using recovery plans, you can also test failovers as well as failback over to the original datacenter.

You also need the ability to easily monitor jobs relating to Hyper-V Recovery Manager and this is also done right through the Hyper-V Recovery Manager console as shown in the following graphic:



When should you consider Hyper-V Recovery Manager? Organizations that are deploying private clouds built on Hyper-V managed by System Center VMM which want ease of use in simplifying replication, failover orchestration, and monitoring of their recovery plans should look at deploying Hyper-V Recovery Manager. You will be amazed by how easy and quickly you can have this solution up and running! I view Hyper-V Recovery Manager as a game changer, the perfect mirroring of onsite services managed by the customer (clouds and virtual machines) managed by a public cloud-based service.

Robert Nottoli
Technology Solution Prof., US - MidWest - STU Core Infra

Learn more

For general information about Windows Azure Hyper-V Recovery Manager and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/recovery-manager/*.

For more detailed information on what Hyper-V Recovery Manager is and how to configure it, see *http://www.windowsazure.com/en-us/documentation/services/recovery-services/*.

For a blog detailing how Hyper-V Recovery Manager works in the context of "in the cloud," see *http://blogs.technet.com/b/in_the_cloud/archive/2013/08/14/what-s-new-in-2012-r2-cloud-integrated-disaster-recovery.aspx*.

Cache

Applications, whether they're running on-premises or in the cloud, tend to access the same data over and over again. One way to boost the performance of applications is to keep data closer to the application in order to minimize delay when the application needs to retrieve that data again. In addition to the Windows Azure Content Delivery Network (CDN) described earlier in Chapter 1, "Understanding Windows Azure,", Windows Azure can als uses several methods for in-memory caching of data in order to ensure users experience the best performance from their cloud-based applications.

To learn more about Windows Azure Cache and the new features currently in preview for this service at the time of writing, let's listen to one of our experts inside Microsoft.

Understanding Windows Azure Caching

Caching in Windows Azure can be divided into the following categories:

- **In-Role Caching**
- **Cache Service (Preview)**
- **Shared Caching (Deprecated)**

Let's examine each of these in turn.

> ***IMPORTANT*** **For any of these caching methodologies, please ensure you have performed rigorous cache capacity planning. Improper capacity planning may result in a throttling situation—one of the most common issues which users run into while using Windows Azure Caching.**

Part 1: In-Role Caching

In-Role Caching (also known as Role-based Caching) was introduced in Win dows Azure SDK version 1.7 as a Preview feature and went live in Windows Azure SDK version 1.8. Since its release, In-Role Caching has become the first choice for Windows Azure customers. In this topology, cached data is stored in the same memory of your Windows Azure role instances. Any other Windows Azure role, within the same deployment, can access the cached data.

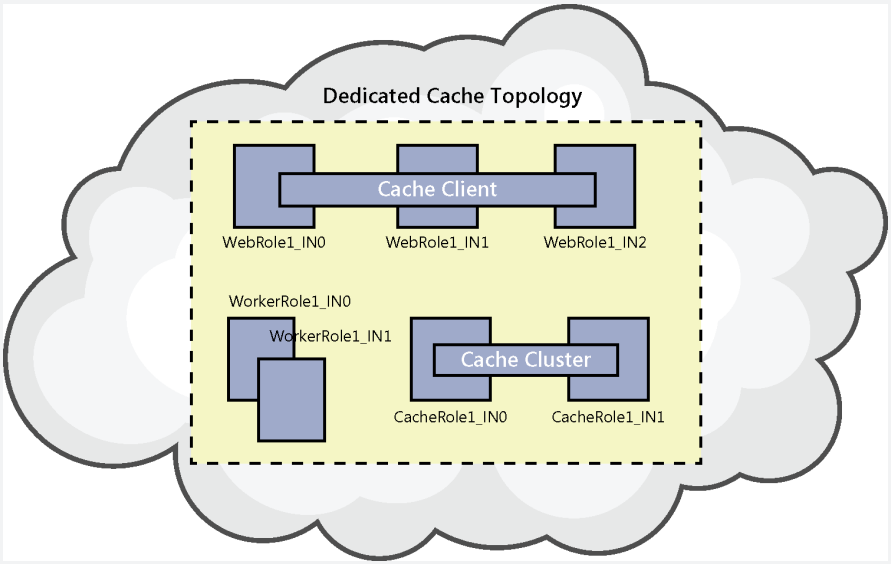
In-Role Caching can be further divided into two categories based on the deployment topologies:

- Dedicated Caching
- Co-located Caching

Let's look at each of these next.

Part 1a: Dedicated Caching

In this topology, you create a separate Worker role to host the cached data. As the name implies, this separate Worker role is dedicated solely for cached data. No other application code or service runs on that role (of course, except the basic Operation System services). Since the Worker role is just used to facilitate cache, theoretically all the available memory on the VM is used to host the cached data. Cached data stays on the VM that runs the Worker role and is accessed by client roles.



The diagram above shows a typical Dedicated Cache architecture. The application contains a Web role (with three instances) and a Worker role (with two instances). The Web role hosts the application which consumes the cache. Since this is a dedicated topology, there is a separate Cache Worker role (two instances) which hosts the cached data.

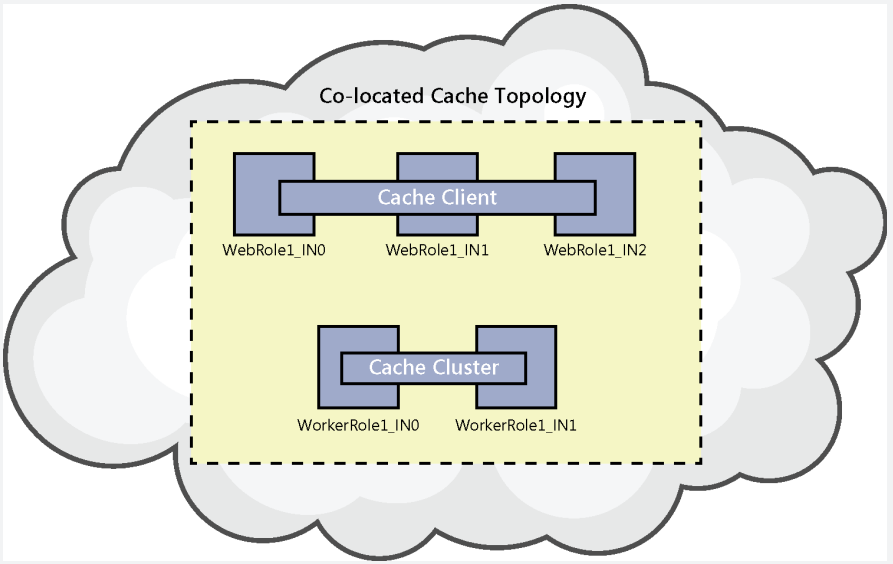
Since this role is essentially a Worker role, you can leverage features like scaling-in/scaling-out, choosing the size of the VM, and rebooting/recycling operations, etc. During the peak business times (for example, during the holiday period) you can scale-out your caching Worker role and increase the number of instances by simply

moving the slider available on the Windows Azure Management Portal. Similarly, when you are not expecting a very high load, you can scale-in the cache roles and decrease the number of instances. Please note that scaling operations may incur data loss.

Part 1b: Co-located Caching

In this topology, you do not create any separate role for cached data. Instead, you reserve a certain portion of memory of VM for the cached data where a caching-enabled role would run. A cache-enabled role can be a Web/Worker role which is already hosting your application code.

For example, you can reserve 30 percent of memory, called Cache Size, of your Web role to host the cached data. This cached data can be accessed by any role within the same deployment, similar to dedicated caching.



The diagram above shows a co-located cache topology. Caching is enabled on the Worker role. So the Worker role will also act as the cache cluster and will host the cached data which will be spread across its two instances.

For example, your cloud project has a Worker role with three instances. If you enable co-located caching on this Worker role and specify the cache size as 30 percent. Then for each of these three instances, 30 percent of total memory will be reserved for cache data. Let's say the Worker role size is small (1.75 GB memory), then total memory available for cache service would be 3*(1.75*30%).

To scale-out/scale-in, you would simply need to increase or decrease the number of role instances. Or you can also increase the size of role, which will effectively increase the amount of memory available for cached data. You can choose either of these methods or a combination of these based on whichever is more suitable for you.

Also, the Cache service is designed to occupy the specified memory as soon as possible. It doesn't wait for the memory to be allocated when the need arises. As a result, you may see that even if you have not put lot of data in cache, the memory consumption by Cache service appears to be increasing as soon as the service starts. As long as the memory usage is stabilizing eventually, this is should be okay.

> NOTE: The total memory consumed by Cache service will be higher than what you specify. The amount of memory you specify is dedicated only for user objects. Apart from that, there will be some operational overhead to manage these objects.

In-Role Caching concepts under the hood

There are two underlying concepts for In-Role Caching which you should be aware of:

- Cache cluster
- Cache client

Let's talk about cache cluster first. In simple terms, cache cluster is responsible for managing your cached data. For either of the caching methods (Dedicated or Co-located) you may have multiple instances hosting the cached data. When such a role that has caching enabled runs on multiple instances, a cache cluster is formed

Cache cluster puts a layer of abstraction in front of these roles hosting your cached data. This enables your application to consume cache data without being bothered about which instances it is actually hosted on. Cache cluster also manages the scaling operations you may perform. In a case where high availability is turned on, cache cluster will perform data replication for you.

> Now let's talk about cache client. Any application which is consuming or storing cached data can be called a cache client. For Role-based Caching (Dedicated or Co-located), cache client must be a part of the same deployment as cache cluster.

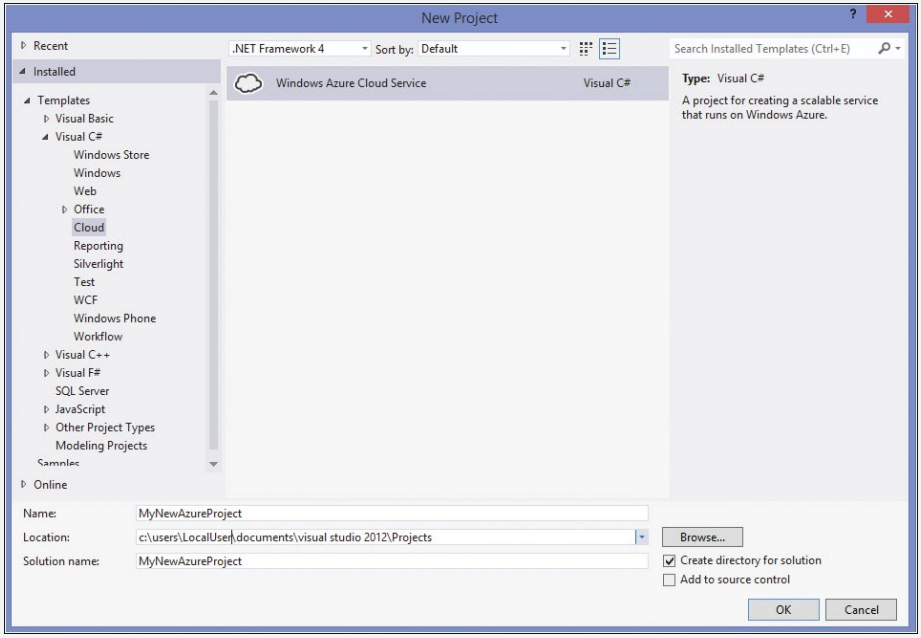
Configuring In-Role Caching for your cloud service

You can configure any of these caching topologies in your cloud service through Microsoft Visual Studio. Cache configuration consists of two parts:

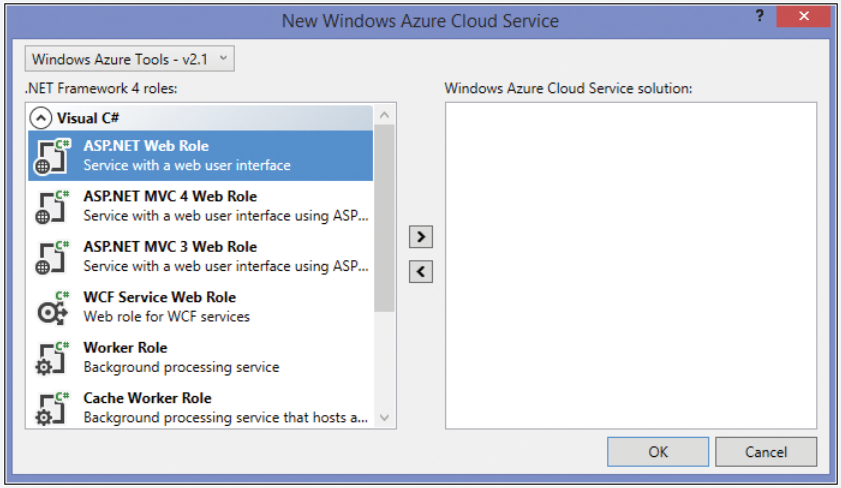
1. Configuring the cache cluster
2. Configuring the cache client

Let's create a new Windows Azure project to see how the cache cluster and cache client can be configured.

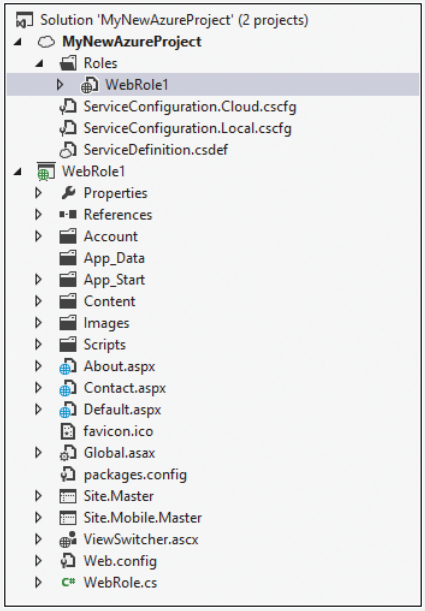
1. Open Visual Studio (2010/2012), click New Project and expand Templates. Choose Cloud project under the language of your choice—Visual C# or Visual Basic. Give it an appropriate name, MyNewAzureProject in this case, and click OK.



2. In the next window, you can add the type of role you want based on your application. For this sample, you will add a ASP.Net Web Role. Select the ASP.Net Web Role from the left pane and click the > button to include it, then press OK.



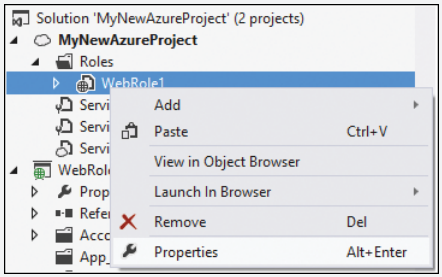
3. In the Solution Explorer, you should see two projects in the solution. One is your role project based on the type of role you added in the previous step and the other one is your cloud project.



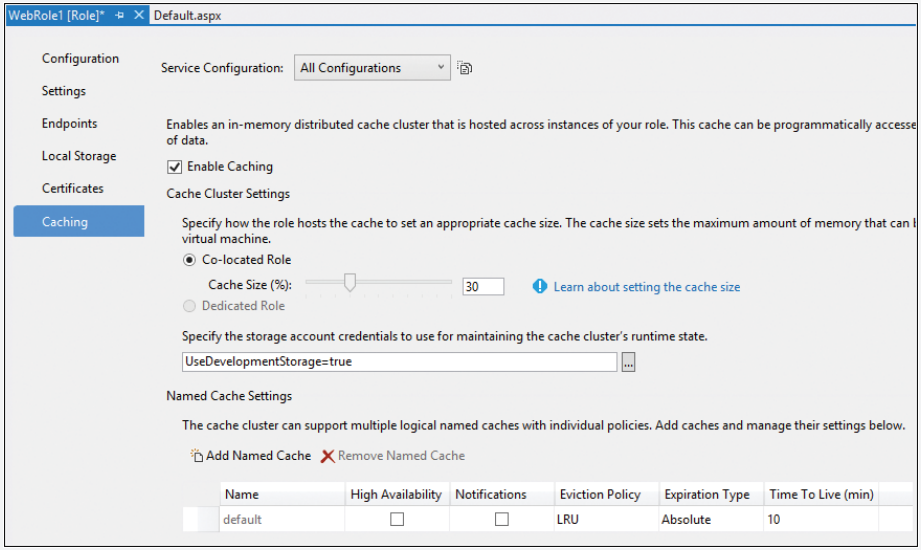
Now the project is ready for caching configuration.

Configuring the cache cluster

1. In the Solution Explorer, go to the cloud project (not the role project), which is MyNewAjureProject in this case. Expand Roles, right-click WebRole1 and go to Properties.



2. In the Properties window, go to Caching. Check the Enable Caching check box.



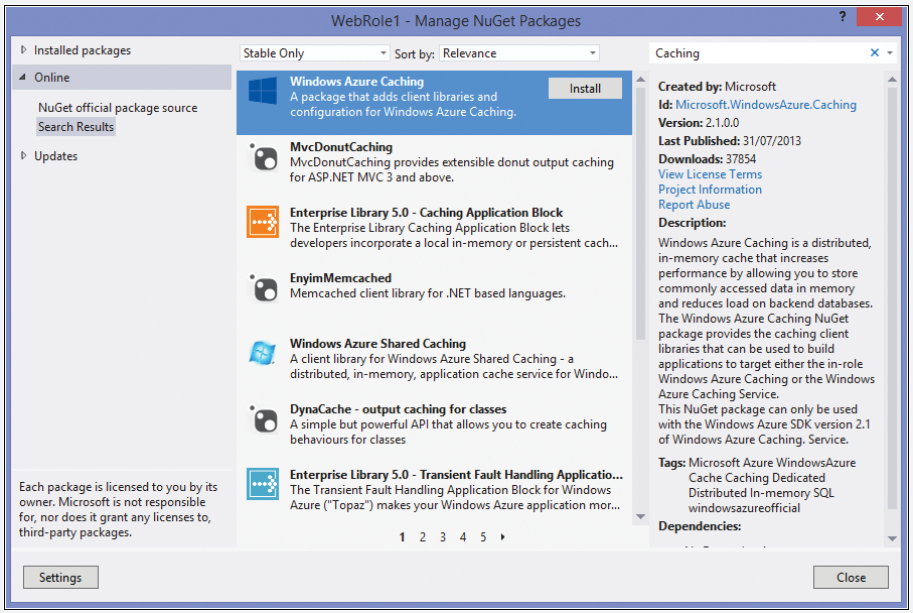
You can select the appropriate Cache Size, configure Storage Account, and also configure Named Cache if required. With this, the cache cluster is now configured.

> ***NOTE*** This step will enable Co-located Caching. If you want Dedicated Caching, then you would need to add a Cache Worker role the way you added an ASP.Net Web role in Step 2 earlier. Once you do that, the Dedicated Caching radio button will be enabled and it will become the default topology for this project.

Configuring the cache client

To configure cache client, you would need to install the Windows Azure Caching NuGet Package at your role project

1. In the Solution Explorer, right-click WebRole1, go to Properties, select Manage NuGet Packages.
2. In the Mange NuGet Packages window, ensure NuGet Official Package Source is selected, type caching (or any similar keyword) in the Search box. Select the Windows Azure Caching package, click Install. Close the window once the package is successfully installed.



The NuGet package will add the required assembly references and appropriate configuration elements in the configuration file (web.config or app.config depending on the Role type).

3. Go to the web.config (or app.config in case of Worker role) file in Visual Studio, find the dataCacheClient element, and replace [cache role name or Service Endpoint] with the name of the role that hosts the cache cluster (WebRole1 in our case), as shown here:

```
<dataCacheClients>
  <dataCacheClient name="default">
    <!--To use the in-role flavor of Windows Azure Caching, set
identifier to be the cache cluster role name -->
    <!--To use the Windows Azure Caching Service, set identifier to
```

```
be the endpoint of the cache cluster -->
      <autoDiscover isEnabled="true" identifier="[Cache role name or
Service Endpoint]" />
      <!--<localCache isEnabled="true" sync="TimeoutBased"
objectCount="100000" ttlValue="300" />-->
      <!--Use this section to specify security settings for connecting
to your cache. This section is not required if your cache is hosted on
a role that is a part of your cloud service. -->
      <!--<securityProperties mode="Message" sslEnabled="false">
        <messageSecurity authorizationInfo="[Authentication Key]" />
      </securityProperties>-->
    </dataCacheClient>
  </dataCacheClients>
```

4. Save the changes and build the project.

These steps are described in detail on the Windows Azure website in the article "How to Use Windows Azure Caching," which can be found at *http://www.windowsazure.com/en-us/develop/net/how-to-guides/cache/*.

General guidelines concerning In-Role Caching

Both of these methods are fully supported by Visual Studio and have Emulator integration. So you can easily, simulate the cache scenario in your local development environment. This greatly simplifies application development and debugging processes. In the development environment, similar to Compute Emulator and Storage Emulator, there will be a Cache Emulator service which will simulate the caching scenario for you.

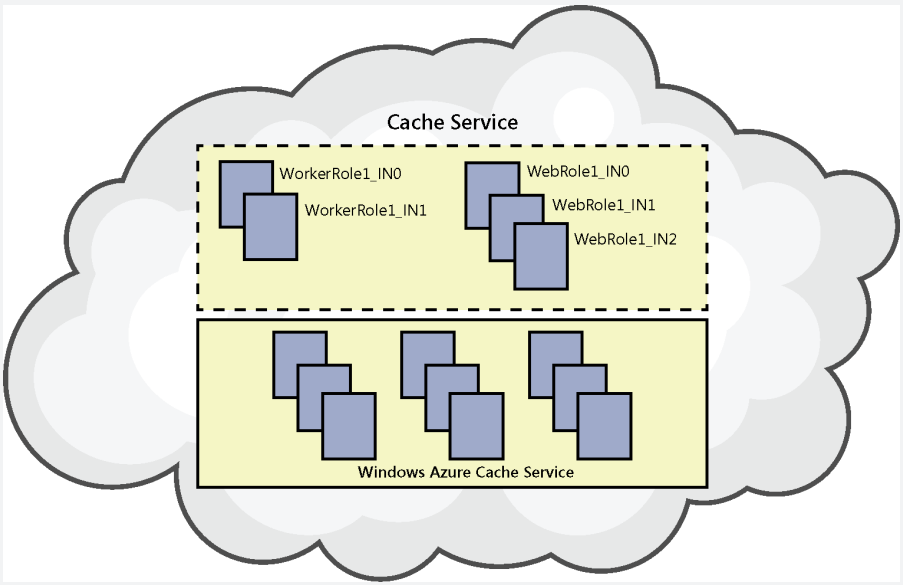
The percent memory specified is applicable only in the Windows Azure environment and not in a development environment. Cache Emulator is designed to consume 16 percent of available memory (and some overhead). You cannot override this behavior.

While you are decreasing the number of instances, it is recommended that you reduce by not more than three instances at a time. Otherwise, your cache cluster can become unstable. Suppose you want to scale-in the cache roles from seven instances to two instances; then you should first reduce three instances (from seven to four), let the roles stabilize, and then reduce further from four instances to two instances. The reason for this lies in the working of cache cluster. During the scale-in operation (reducing the number of cache instances), cache cluster attempts to move the data from scaled-in instances (instances which will be taken down) and redistribute it to other instances. This operation may not work correctly if you attempt to reduce a large number of instances in one go.

Part 2: Cache Service (Preview)

Windows Azure Cache Service is the latest offering for caching in Windows Azure. It was preview released on September 3, 2013. This is a high performance, distributed, in-memory, and dedicated infrastructure based Cache Service. It is designed to work with a spectrum of applications like Windows Azure Cloud Services, IaaS VMs, and Windows Azure Web Sites. Support for more services is expected to be added in the near future.

Cache Service is very easy to create and provides full feature parity with other cache offerings. Like Shared Caching, it hosts your cached data on a pool of servers which are managed by Microsoft. But unlike Shared Caching, this is a dedicated infrastructure based service. Also, there are no quotas/throttling based on the usage.



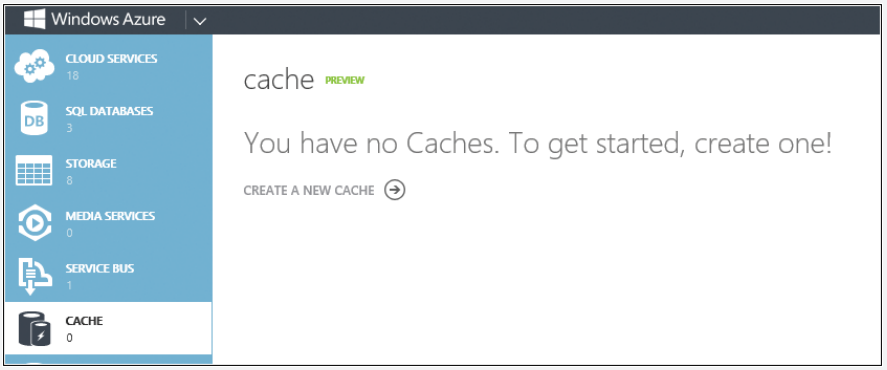
In the above diagram, with Cache Service being configured, cached data is hosted on a pool of servers which are beyond application deployment boundaries. This hosting infrastructure is managed by Microsoft.

Configuring Cache Service

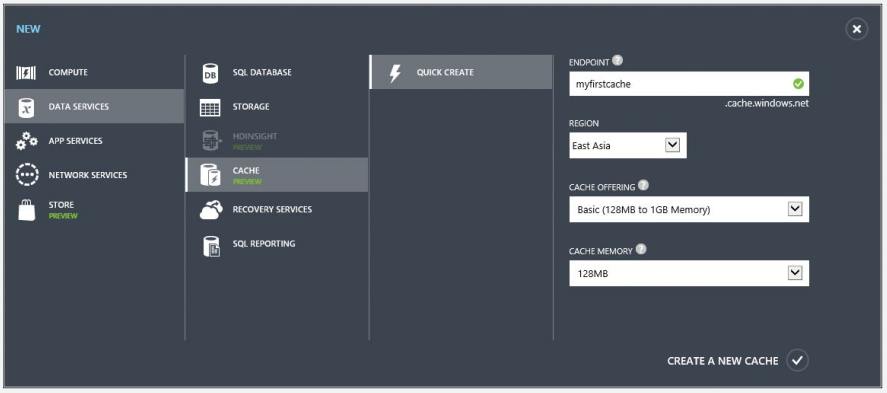
Cache Service can be configured and managed through the Windows Azure Portal (*https://manage.windowsazure.com*). You can choose a suitable offering (Basic, Standard, or Premium) and accordingly choose the appropriate cache size (from 128 MB to 150 GB). You can then create a Cache Service end point in the datacenter of your choice. This service end point, along with the access key, is used to configure client applications to start using Cache Service.

Here are the steps to configure Cache Service in the Windows Azure Portal:

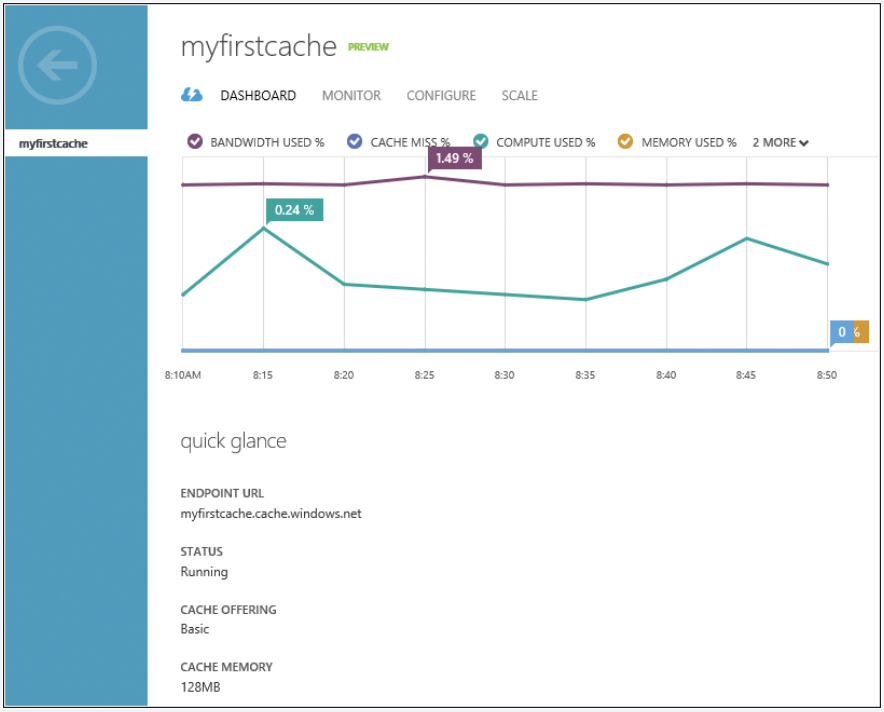
1. In the Management Portal, you should see a Cache tab in the left panel which lets you create/manage Cache Service.



2. To create a Cache Service click New, select Data Services, Cache Preview, and select Quick Create (or you can also click on Create New to do the same). Give a name to Cache Endpoint, choose the Data Center Region, Cache Offering (Basic, Standard, or Premium) and Cache Memory size.



3. Click Create A New Cache and wait a few minutes for it to provision the cache. Once provisioned, you should see a new cache endpoint created and it should be in running state.



4. The newly created Cache Service is now ready to be used by your application. If you need to further customize the features, Windows Azure Portal provides various option tabs for this.

Here is a brief description of what those tabs allow you to do:

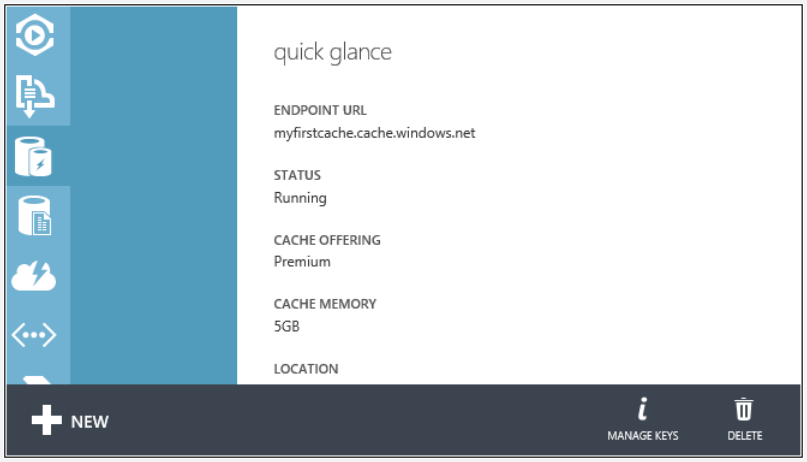
- **Dashboard** Provides details about the cache endpoint along with various usage metrics (see the previous graphic above)
- **Monitor** Enables you to add custom metrics (up to 12) as per your monitoring needs.
- **Configure** This tab allows you to customize features like Expiry Policy, Eviction, and High Availability. The feature set on this tab varies according to the Offering type you have selected (Basic, Standard, or Premium) as follows:
  - **Basic Cache Offering** Expiry Policy, Expiration Time, and Eviction Policy
  - **Standard Cache Offering** Expiry Policy, Expiration Time, Notification, Eviction Policy, and Named Cache
  - **Premium Cache Offering** Expiry Policy, Expiration Time, Notification, High Availability, Eviction Policy, and Named Cache
- **Scale** Using this tab you can modify the Cache Offering and the Cache Memory (size). Please note that this operation results in loss of data.

Once you have created a new Cache Service, now you need to configure your client application to start using this. The steps to configure the client are very much similar to what you do in case of In-Role Caching. We can divide these steps in two parts:

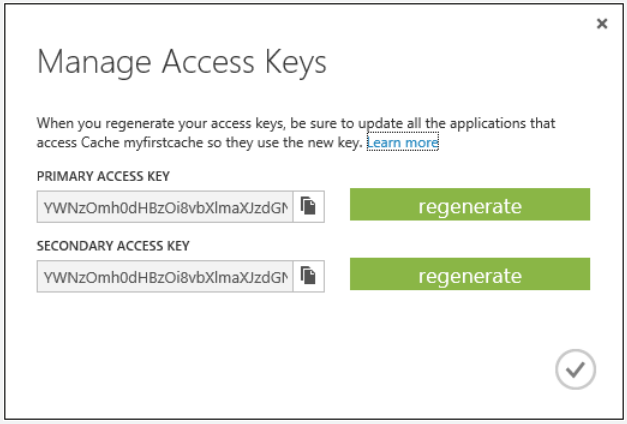
1. Gather Cache Endpoint Access Information
2. Configure Client Application

Gather Cache Endpoint Access Information

Client application would need Endpoint URL, and Access Key to be able to access the cache endpoint. This information can be found on the Windows Azure Portal, at the Dashboard tab for the cache endpoint which you have just created. The End Point URL can be found in the quick glance section. You can copy the URL for later usage:

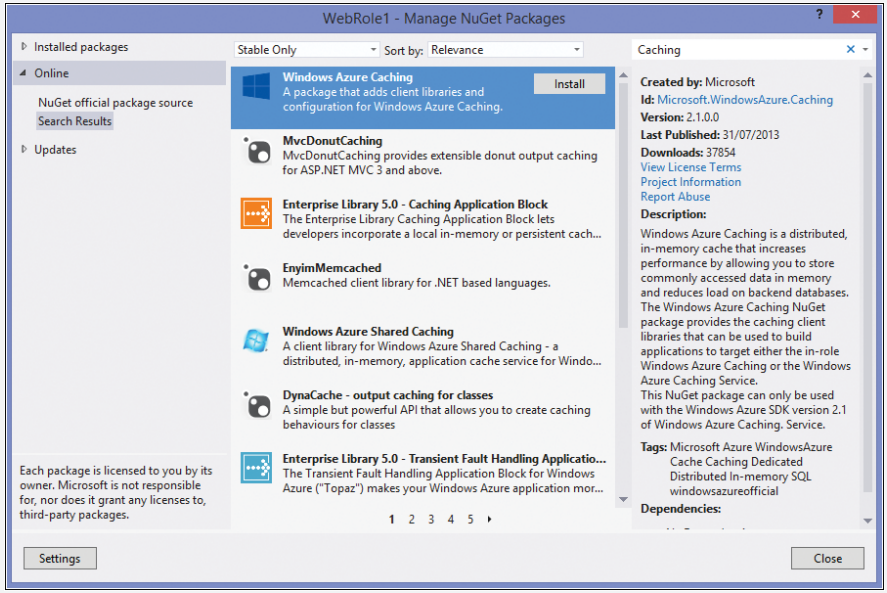


To find the Access Key, click the Manage Keys button at the bottom of the page. You can copy any one of the access keys for later usage:



Configure Client Application

1. In Microsoft Visual Studio 2010/2013, open your application project and go to Solution Explorer, right-click project from where you want to access cache, go to Properties, select Manage NuGet Packages.
2. In the Manage NuGet Packages window, ensure NuGet official package source is selected, type Caching (or any similar keyword) in the Search box. Select the Windows Azure Caching package, click Install. Close the window once the package is successfully installed:



The NuGet package will add required assembly references and appropriate configuration elements in the configuration file (web.config or app.config depends on the application type).

3. Go to the web.config (or app.config) file in Visual Studio, find the dataCacheClient element and replace [cache role name or Service Endpoint] with the Cache Service Endpoint URL and [Authentication Key] with the Access Key. Both these data points you have copied from the Azure Portal in preview steps:

```
<dataCacheClients>
  <dataCacheClient name="default">
    <!--To use the in-role flavor of Windows Azure Caching, set
identifier to be the cache cluster role name -->
    <!--To use the Windows Azure Caching Service, set identifier to
be the endpoint of the cache cluster -->
    <autoDiscover isEnabled="true" identifier="[Cache role name or
Service Endpoint]" />
    <!--<localCache isEnabled="true" sync="TimeoutBased"
objectCount="100000" ttlValue="300" />-->
    <!--Use this section to specify security settings for connecting
to your cache. This section is not required if your cache is hosted on
a role that is a part of your cloud service. -->
    <securityProperties mode="Message" sslEnabled="false">
      <messageSecurity authorizationInfo="[Authentication Key]" />
    </securityProperties>
  </dataCacheClient>
</dataCacheClients>
```

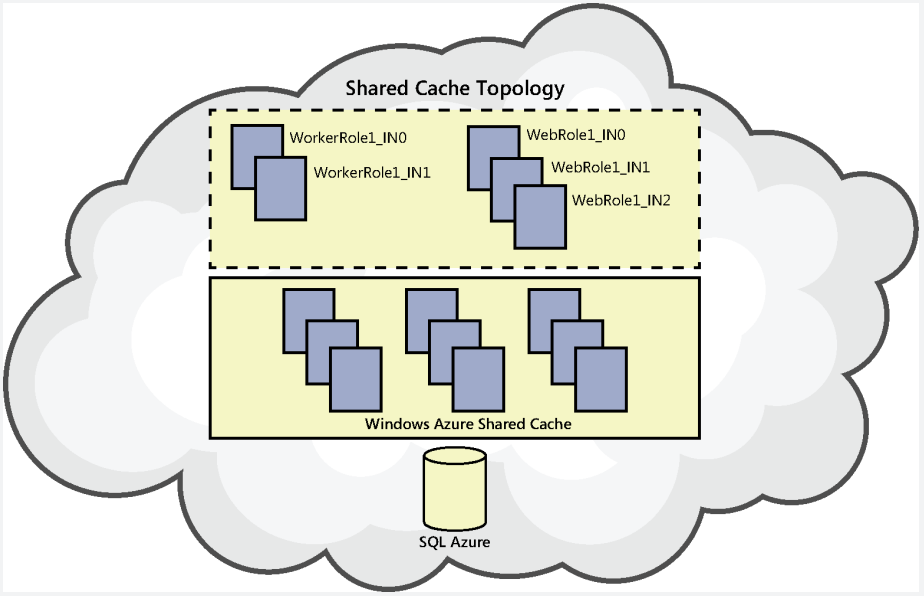
4. Save the changes and build the project. Your client application is now ready to start using the Cache Service.

More information around this can be found at the following MSDN documentation listed at *http://www.windowsazure.com/en-us/documentation/services/cache/*.

Part 3: Shared Caching (Deprecated)

Shared Caching, also known as Distributed Azure Caching, Azure AppFabric Caching, or Multitenant Caching, is the legacy way of caching in Windows Azure. As of September 3, 2013, Shared Caching is deprecated and will be retired no later than August 29, 2014 as the Silverlight Portal used to manage Shared Caching will be decommissioned on March 31, 2014. Shared Caching service users who have not migrated to the new Windows Azure Cache will receive communications on how to manage their shared caches if they choose to continue using this service after March 31, 2014.

In the shared mode of caching, your cached data is stored on a pool of a server which is managed by Microsoft. As the name implies, these cache servers are multitenant. They host multiple, isolated caches for multiple users. Shared Caching does not have any limitation of accessibility only within the same deployment. On the contrary, since you get a namespace for Shared Caching, virtually any application possessing Internet connectivity and an access token can connect to the shared cache.



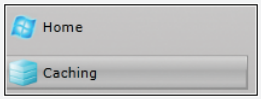
The diagram above shows that with Shared Caching being enabled, cached data is hosted on Windows Azure Cache Servers managed by Microsoft. Unlike In-Role Caching, these cache servers are beyond your subscription boundaries and thus are beyond your control. You just subscribe to one of the shared cache offerings and accordingly, you are allocated memory for your cached data. Along with the memory limit, there are other usage quota limitations to ensure fair usage of resources by all the users (tenants). These limitations could be based on the number of transactions being made against the cache, the total bandwidth being consumed, or the number of concurrent connections used. If you exceed the limitations enforced by the service, your application receives an exception that specifies the exceeded quota limit.

> ***NOTE*** Shared Caching is designed to be used with Windows Azure applications hosted in the cloud. This achieves the best throughput at the lowest latency. While is it possible to consume cached data from an on-premises application, it is not really a supported scenario.

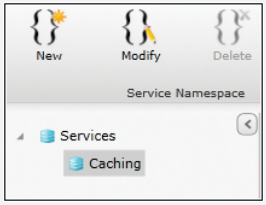
Configuring Shared Caching for your cloud service

To configure Shared Caching, you need to use the previous Windows Azure Management Portal (*https://windows.azure.com*). This is because as of Aug 8, 2013, the previous Windows Azure Management Portal is used only for Windows Azure cache configuration. For managing all other services, you need to use the new portal (*https://manage.windowsazure.com*) which does not support Shared Caching configuration.

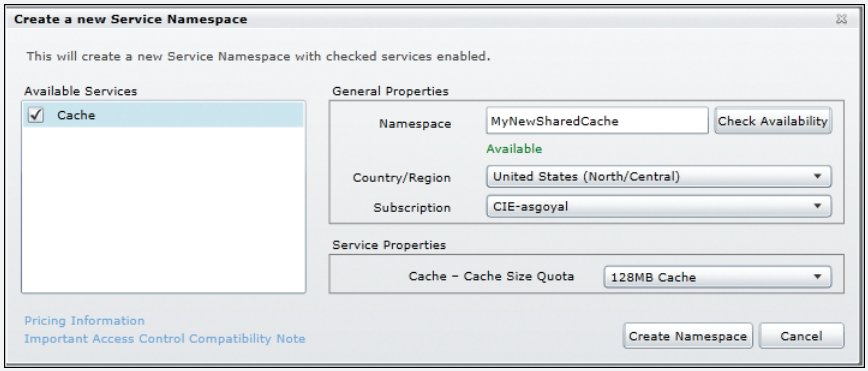
1. Go to the previous Management Portal (*https://manage.windowsazure.com*). You need to have a valid Windows Azure subscription.
2. On the Management Portal home page, at the lower left of this page, click Caching.



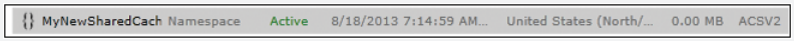
3. Click the New button on the toolbar.



4. In the Create A New Service Namespace dialog, verify that the Cache checkbox is selected under Available Services.
5. Type a proposed namespace in the Choose A Service Namespace text box, and then click Check Availability. If the message "Available" appears under the text box, then the namespace name is available, and you can continue.
6. Populate rest of the fields as appropriate (Country/Region, Subscription, Cache Size Quota).
7. Click Create Namespace.



8. **Once the provisioning is done, your Cache Namespace will be shown as Active.**



Further details about these steps and information on client configuration and using Shared Caching in your application can be found in the article, "Developing for Windows Azure Shared Caching," which can be found at *http://msdn.microsoft.com/en-us/library/windowsazure/gg278342.aspx*.

Ashish Goyal
Support Escalation Engineer, Azure Cloud Integration Engineering

Learn more

For general information about Windows Azure Cache and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/cache/*.

For more detailed information on what Windows Azure Cache is and how to get started using it, see the following Planning Guides on MSDN:

- "Capacity Planning for Windows Azure Cache Service (Preview)," which can be found at *http://msdn.microsoft.com/en-us/library/windowsazure/dn386139.aspx*
- "Cache Offerings for Windows Azure Cache Service (Preview)," which can be found at *http://msdn.microsoft.com/en-us/library/windowsazure/dn386114.aspx*

CHAPTER 5

Windows Azure app services

Running applications is the heart of what Windows Azure is all about. To enable running your business applications in the cloud, Windows Azure provides services for managing user identity, enabling applications to interact with each other, push out notifications to users, deliver compelling media experiences, and more. This chapter examines the Windows Azure app services available at the time of writing.

Windows Azure AD

Managing identity is foundational to the security of any application that users can interact with. Windows Azure Active Directory (Windows Azure AD) is a cloud-based version of Active Directory that provides a subset of the functionality of the familiar Active Directory Domain Services (AD DS) that is used by so many businesses around the world as their identity and access control solution.

Windows Azure AD provides a cloud-based identity provider that can integrate into your on-premises AD DS deployments. Windows Azure AD is also the identity solution used by Microsoft Online Services such as Windows Azure, Microsoft Office 365, Dynamics CRM Online, and Windows Intune. Because of these things and because of its ability to integrate with web identity providers like Microsoft Account and popular third-party providers like Google, Yahoo!, and Facebook, Windows Azure AD can provide single sign-on for users across Microsoft Online Services, third-party cloud services, and applications built on Windows Azure.

Deploying Active Directory using Windows Azure Virtual Machines

Windows Azure also supports other ways of deploying Active Directory in the cloud. For example, you can deploy an entire Active Directory forest in a self-contained manner using Windows Azure Virtual Machines. Another identity and access control scenario that Windows Azure supports is a hybrid deployment where an organization's domain controllers are partly deployed on-premises and partly deployed on Windows Azure Virtual Machines. For more information on these types of deployment scenario, see "Guidelines for Deploying Windows Server Active Directory on Windows Azure Virtual Machines," which can be found at *http://msdn.microsoft.com/en-us/library/windowsazure/jj156090.aspx*.

Using Windows Azure AD

When you log on to the Windows Azure Management Portal for the first time and select the Active Directory tab on the left, an item called the Default Directory is displayed as Active for your subscription, as shown in Figure 5-1.

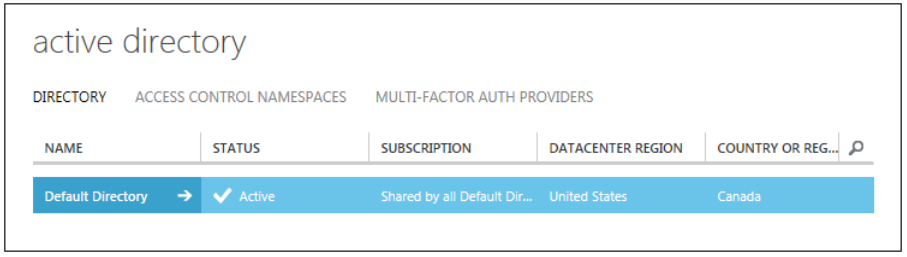


FIGURE 5-1 The Default Directory in Windows Azure AD is marked as Active.

If you click on Default Directory, a page similar to Figure 5-2 opens.

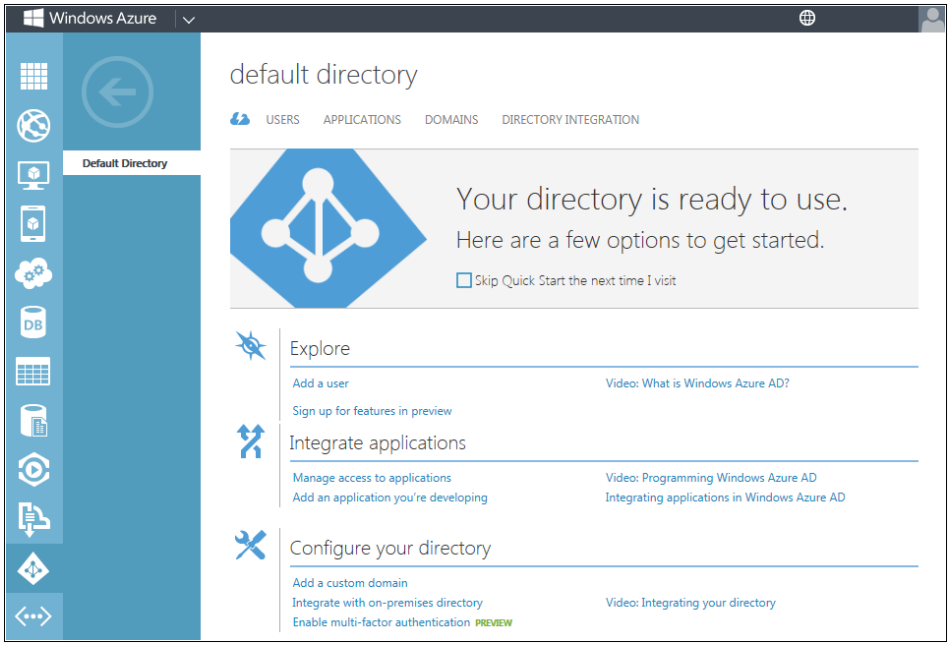


FIGURE 5-2 The tasks you can perform are displayed in the Default Directory.

As you can see from Figure 5-2, you can use this page of the Management Portal to create and manage new user accounts, add applications and manage access to them, and perform other identity management tasks.

Let's now hear from one of our Microsoft insiders to learn in more detail what Windows Azure AD is all about and what your business can do with it.

Securing cloud services using Windows Azure AD

Authentication and authorization is the fundamental requirement of every application that provides useful value to its users in terms of the tools and processes it automates. Until the advent of Windows Azure AD, Windows Azure applications from companies with no on-premises AD infrastructure had to host their own AD instance or use custom security implementation that involved the storage of user attributes and passwords on Windows Azure Table Storage, Azure SQL Database, or numerous other options offered by the Virtual Machines. Windows Azure AD not only relieves developers from managing user identities per application but also helps with the single sign-on across multiple applications that span public/private clouds and in-house as well as vendor applications.

Windows Azure AD versus AD DS

Windows Azure AD is a multitenant cloud-hosted directory service that is designed to store objects, including users, groups, roles, and contacts, and provide authentication and authorization services for applications including Office 365 and applications hosted on Windows Azure and other public and private clouds. Windows Azure AD is different from the familiar AD DS which is a comprehensive LDAP directory that has been the cornerstone of enterprise resources and security principal management since Windows Server 2000. AD DS can be installed on on-premises physical Windows servers or on Windows Server virtual machines offered by various public clouds including Windows Azure.

AD DS stores resources (for example, printers, file shares, conference rooms) and security principals (for example, users, computers, groups) and organizes them into a hierarchical directory using concepts like organizational units, forests, and trees and domains. AD DS also implements group policies, replication services, Kerberos, and global catalog services for enabling the operational aspects of the information the directory manages.

Windows Azure AD offers a subset of functionality that is included with AD DS; Windows Azure AD stores users, groups, contacts, tenant information, licensing, roles, and service principals with the sole purpose of enabling user authentication and information queries through Graph API. To enable hybrid cloud scenarios where a large enterprise may have applications distributed across on-premises (secured by AD DS) and the cloud (secured by Windows Azure AD), a subset of the AD DS information (users, groups, and contacts) will be synchronized with Windows Azure AD on a continual basis.

Industry standard protocols

Windows Azure AD implements industry standard WS-Trust, WS-Federation, and SAML-2 standards for platform agnostic integration of authentication services. As a result, Windows Azure AD can be used in securing applications from Microsoft as well as non-Microsoft platforms. Windows Azure AD supports both SAML and JWT token formats that allow the parsing and processing of the Windows Azure AD authentication tokens across a diverse set of applications and services in an interoperable manner. Windows Azure AD is also an OAUTH 2.0 provider that allows the securing of rich client, web app, and web services using JWT tokens. In addition to authentication, Windows Azure AD publishes Graph API over REST with ODATA and JSON payloads for allowing applications to read and write the directory objects. These objects include user, contact, group memberships, and role information which can be used in enforcing application specific authorization policies. Reading of the directory information is only allowed by the applications with explicit consent by the owners as is described in OAUTH 2.0 specification.

Synchronization of AD DS with Windows Azure AD

In a hybrid directory setting, Windows Azure AD is a projection of the on-premises active directory into the cloud and requires continual synchronization throughout the duration of the projection lifetime. DirSync is the tool used in implementing the synchronization process which is installed close to AD DS. DirSync server polls for the object changes and uploads them to the cloud—every three hours. Previously, DirSync could not synchronize password hashes and hence the only option for authenticating users in a hybrid directory (a combination of AD DS and Windows Azure AD) scenario is through federated authentication. The latest release of DirSync supports synchronization of hashes computed from password hashes; given the sensitivity of the password changes, the password synchronization happens in near real-time.

The difference between Federated Authentication and Managed Authentication is that in the federated setting, the user authentication for all users happens through ADFS attached to the on-premises AD DS while the user validation for non-federated users happens at Windows Azure AD with no ADFS traversal. At the time of this writing, the self-contained user authentication process supported by WWAD enabled by the password synchronization is referred to as Managed Authentication.

Password synchronization can only be enabled for the entire namespace and can't be done at the individual user level. Hybrid directory environments can be set up in such a way that some namespaces can be federated while others can support Managed Authentication through password synchronization. One caveat with password synchronization is that this is not a true single sign-on; when a user accesses an application secured with Windows Azure AD, even though the user has already been authenticated against AD DS (for example, during the workstation log on), the user will be challenged again for credentials.

Graph API

Windows Azure AD publishes access to the objects stored within the directory through a set of REST APIs. The API covers CRUD operations on user and group objects and read operations on role and contact objects at the time of this writing. The REST API is secured through OAuth 2 and hence the application has to be granted access either by the administrator implicitly or explicitly by the users. In either case the application will make a request for the authentication token, in this case the format is JWT (JSON Web Token), using application id (aka client id) and the secret key. Once the application is authenticated, a token is returned with TTL which will be used in making calls to the directory for accessing the user, group, role, and contact information. In addition to absolute queries, Graph API also supports differential queries which can save time and bandwidth if the app needs to know the changes since the last query.

Windows Azure AD implementation scenarios

Windows Azure AD (in standalone or federated mode) supports a variety of authentication and authorization scenarios, including the following:

- A web application, device app, a background process/services accessing a set of web services
- A web application, a web service, device app, a background process/service needs to authenticate and authorize users

There may be cases where the graph information may not be granular enough for role-based security implementation in applications. Under such circumstances, applications may be able to maintain fine-grained role mappings to the canonical user attribute (for example, user id or primary email address) and enforce application entitlements. This is completely encapsulated within the application and is complimentary to Windows Azure AD graph information.

Given the platform-agnostic protocol support in Windows Azure AD, it can be used for securing applications written for .NET as well as numerous other runtime platforms. Active Directory Authentication Library, which is in beta at the time of this writing, helps .NET programmers to acquire authentication tokens from Windows Azure AD and AD DS without writing any protocol specific code.

Summary

Windows Azure Active Directory is a cloud-hosted multitenant directory service that supports industry standard authentication protocols and token formats so that it is interoperable with Microsoft as well as non-Microsoft application platforms. Windows Azure AD can be federated with on-premises AD DS to create a single sign-on infrastructure so that the authentication context will seamlessly flow between applications hosted across on-premises and multiple clouds. Windows Azure AD can also operate in a standalone setting in addition to federated hybrid directory mode.

Hanu Kommalapati
Senior Technical Director, WW DPE

Learn more

For general information about Windows Azure Active Directory and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/active-directory/*.

For more detailed information on what Windows Azure Active Directory is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/active-directory/*.

Finally, be sure to check out these TechEd 2013 presentations that cover Windows Azure Active Directory and other identity and access control technologies supported by Windows Azure. These presentations are available for viewing and download from Channel 9 here:

- "Introduction to Windows Azure Active Directory," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B309*
- "Securing Cloud Line-of-Business and SaaS Web Applications Using Windows Azure Active Directory," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B306*
- "Securing Rich Client Applications Using OAuth 2.0 and Windows Active Directory," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B307*
- "Running your Active Directory in Windows Azure Virtual Machines," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/MDC-B300*
- "Deep Dive into the Windows Azure Active Directory Graph API: Data Model, Schema, Query, and More," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B308*

Multi-Factor Authentication

Windows Azure Multi-Factor Authentication is a new Windows Azure identity service that is currently in preview at the time of writing. This service can be used to provide an extra layer of authentication in addition to user account credentials by requiring them to use their phone or another device together with their password whenever they want to sign in. Multi-Factor Authentication can be helpful when you need stronger access control for employees, customers, or partners. Multi-Factor Authentication can be especially useful for enabling organizations to achieve regular compliance in certain industry verticals.

Multi-Factor Authentication can be used to control access to both on-premises and cloud-based applications for your business. For example, you can use Multi-Factor Authentication to secure access to Windows Azure and to other Microsoft Online Services like Office 365 and Dynamics CRM Online. You can also use it to secure access to third-party cloud services that integrate Windows Azure AD.

NOTE **Multi-Factor Authentication also works with other identity stores. For more details, see *http://technet.microsoft.com/en-us/library/dn249479.aspx*.**

Using Multi-Factor Authentication

You can use the Windows Azure Management Portal to create a new Multi-Factor Authentication provider in Windows Azure (see Figure 5-3). You can create a new Multi-Factor Authentication identity provider with one of two usage models: Per Enabled User or Per Authentication, and you can choose to link to a particular directory or not.

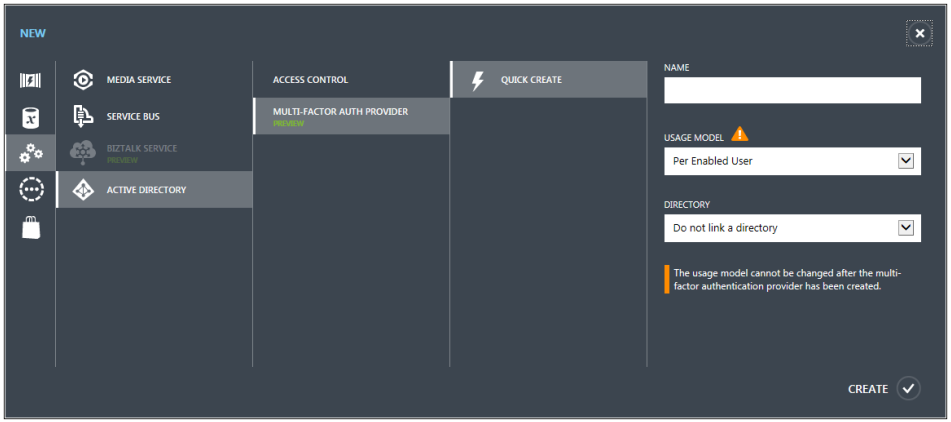


FIGURE 5-3 You can use the Quick Create option to create a new Multi-Factor Authentication provider.

For example, once you've created a Multi-Factor Authentication provider using the Per Enabled User model, you can then configure the properties of users in Windows Azure AD to require Multi-Factor Authentication the next time they log on (see Figure 5-4).

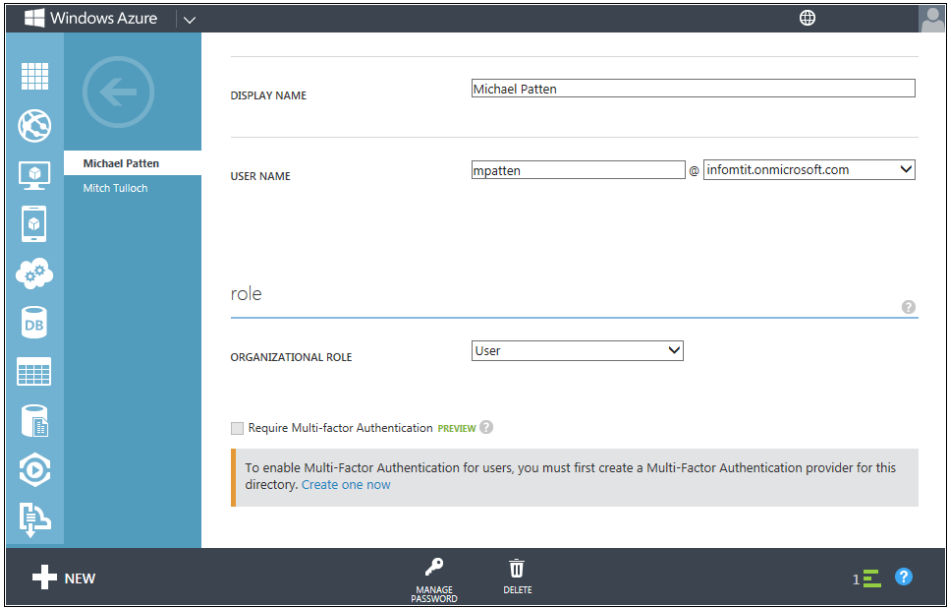


FIGURE 5-4 Enabling Multi-Factor Authentication for a user in Windows Azure AD.

Learn more

For general information about Windows Azure Multi-Factor Authentication and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/multi-factor-authentication/.*

For more detailed information on what Windows Azure Multi-Factor Authentication is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/multi-factor-authentication/.*

Messaging

Business applications are often multitier in nature, and code running in each tier needs to be able to communicate with code in other tiers in a way that is fast, secure, and reliable. For applications deployed in the cloud, this can become a key issue since different application components often run on physical servers in datacenters located in different geographical locations, sometimes even on separate continents.

Windows Azure provides several ways for the different components of a cloud-based application to effectively communicate with one another: Windows Azure Queue, Windows Azure Service Bus, and Windows Azure Notification Hubs.

Windows Azure Queue

Let's say that one piece of application code sends a message to another piece of code. What if the receiving code isn't ready to process the message? In that case, a simple solution is to temporarily store the message in a queue until the receiving code is ready to process it.

For cloud-based applications that can use this kind of approach, Windows Azure Queue can provide exactly what they need. For example, when PHP code running in a Web role needs to communicate with code running in a Worker role of the same cloud service, simple message queuing using Windows Azure Queue enables the Worker role to perform asynchronous processing of the message when the role is ready to do so.

Message queuing has several advantages when building cloud-based applications. For one thing, it's easy to implement. It also scales well, for in the previous example you could easily increase the number of Web role instances or Worker role instances to meet increasing demand.

Windows Azure Service Bus

While simple message queuing is good for one-to-one communications between application components, it doesn't provide a good solution in scenarios where one-to-many communication is needed between components. To address this need, Windows Azure provides another service called Windows Azure Service Bus.

Windows Azure Service Bus supports both basic queuing and publish-and-subscribe forms of messaging. The publish-and-subscribe approach enables one piece of code to send a message on a topic and have multiple other pieces create subscriptions to the topic. Service Bus also includes a relay service that enables application components to securely communicate through firewalls by exchanging messages through an endpoint hosted in the cloud. In Windows Azure terminology, an endpoint is one or more ports by which you can directly access an individual instance of a service. Application code communicates with Windows Azure services by binding to endpoints. Service Bus thus enables Windows Azure applications to communicate with other Windows Azure applications, with applications running on some other cloud platform, or even with applications running outside the cloud.

Windows Azure Notification Hubs

If applications or application components running in the cloud need to engage the user in some way, they need a way to ensure a message sent will be received by the user. Push notifications are a good way of making this happen. But what if an application needs to push out a notification to millions of users around the world?

Windows Azure Notification Hubs was created to address this kind of scenario. Notification Hubs provides cloud-based applications with a method of broadcasting cross-platform push notifications quickly to large numbers of recipients. Notification Hubs can be used with applications running in Windows Azure Virtual Machines, Cloud Services, Web Sites, or Mobile Services. Notification Hubs currently supports push notifications for the following platforms: Windows Phone, Windows Store, iOS and Android.

With Notification Hubs for example, a cloud-based financial application running in Windows Azure can automatically push out updates to a tile on a Windows Phone so the user can always see the latest financial information. Notification Hubs are ideal for cloud-based applications accessed by smartphones and other handheld devices.

Using Windows Azure Messaging Services

As Figure 5-5 shows, when you create a new Service Bus you have four options to choose from:

- You can create new Service Bus queues. When you create a new queue you can configure its maximum size, the time to live (TTL) for messages stored in the queue, the duration of time a message is locked when a receiver is accessing it, and other configuration settings. You can also manage queues you've created, for example by viewing the queue length and last access times for your queues.
- You can create new Service Bus topics. When you create a new topic you can configure its maximum size, the time to live (TTL) until the topic expires, and other configuration settings. You can also manage topics you've created, for example by viewing the subscriptions count and last access times for your topics.

- You can create new Service Bus relays in Service Bus namespaces you've created in Windows Azure. You can also manage relays you've created, for example by configuring credentials.
- You can create new Service Bus notification hubs. You can also manage notification hubs you've created, for example by monitoring registrations and active devices.

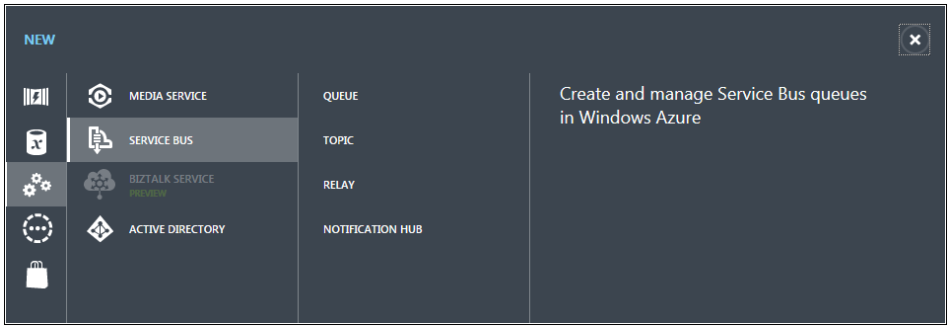


FIGURE 5-5 You can find options for creating new Service Bus items in the Management Portal.

Learn more

For general information about Windows Azure Service Bus and Windows Azure Queues and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/messaging/*.

For more detailed information on what Windows Azure Service Bus is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/service-bus/*.

For general information about Windows Azure Notification Hubs and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/notification-hubs/*.

For more detailed information on what Windows Azure Notification Hubs is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/notification-hubs/*.

Finally, be sure to check out these TechEd 2013 presentations that cover Windows Azure Service Bus and other types of Windows Azure Messaging. These presentations are available for viewing and download from Channel 9:

- "Messaging with Windows Azure Service Bus," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B310*
- "Connected Clients and Continuous Services with Windows Azure Service Bus," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B336*
- "Patterns of Cloud Integration," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/ATC-B220*

BizTalk Services

Windows Azure BizTalk Services is currently in preview at the time of writing. BizTalk Services provides an extensible cloud-based integration service for implementing Business-to-Business (B2B) and Enterprise Application Integration (EAI) capabilities for cloud and hybrid solutions.

BizTalk Services is ideal for building such platforms as cloud-based Electronic Data Interchange (EDI) systems and other kinds of B2B systems. BizTalk Services can connect to HTTP, FTP, SFTP, or REST data sources, and it can be integrated with third-party systems such as those from SAP, Oracle EBS, SQL Server, and PeopleSoft.

Learn more

For general information about Windows Azure BizTalk Services and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/biztalk-services/*.

For more detailed information on what Windows Azure BizTalk Services is and how to get started using it, see *http://www.windowsazure.com/en-us/documentation/services/biztalk-services/*.

Finally, be sure to check out these TechEd 2013 presentations that cover Windows Azure BizTalk Services which are available for viewing and download from Channel 9:

- "Introduction to Windows Azure BizTalk Services," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B313*
- "Extending Windows Azure BizTalk Services," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B315*
- "B2B Collaboration on Windows Azure," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B343*

Media Services

Delivering rich media experiences is a key requirement for many business applications today. Doing this with cloud-based applications presents special challenges in terms of performance, scalability and security.

Windows Azure Media Services is a cloud-based service that lets you build workflows for creating, managing, and distributing media across a wide range of user devices ranging from the Xbox platform to Windows PCs, Windows Phone, MacOS, iOS, and Android devices. Media Services can also be integrated with the tools and processes your business already uses for delivering media experiences to users or customers.

Learn more

For general information about Windows Azure Media Services and for purchase options and pricing details, see *http://www.windowsazure.com/en-us/services/media-services/*.

For more detailed information on what Windows Azure Media Services is and how to get started using it, see *http://www.windowsazure.com/en-us/develop/media-services/*.

Finally, be sure to check out these TechEd 2013 presentations that cover Windows Azure Media Services which are available for viewing and download from Channel 9:

- "Building Rich Media Applications with Windows Azure Media Services," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B318*
- "Building Media Workflows in the Cloud with Windows Azure Media Services," which can be found at *http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013/WAD-B317*

CHAPTER 6

Getting Started with Windows Azure

Now that you're familiar with the Windows Azure platform, you're ready to take the next steps.

Try Windows Azure

There's no better way of finding out about the powerful capabilities of Windows Azure than by trying out the platform. There are several ways you can do this without incurring any cost on your part.

Microsoft is currently offering a free one-month trial of Windows Azure that provides you with $200 of Windows Azure credits you can use for whatever you want. You get full access to the platform with no strings attached, so you can create and run virtual machines, build websites, develop modern apps in the cloud, and lots more. Just sign in with your Microsoft account and fill out the form. You need to provide credit card info, but your credit card won't be billed unless you choose to activate a paid subscription. To sign up, go to *http://www.windowsazure.com* and click Free Trial.

If you're a developer and already have an MSDN subscription for Microsoft Visual Studio Professional, Premium or Ultimate, you get up to $150 Windows Azure credits per month. You can activate your MSDN benefit right now and start building native cloud applications, testing existing applications in virtual machines, or creating hybrid applications that span your datacenter and Windows Azure. To activate these benefits, log on to the MSDN Subscriptions site at *http://msdn.microsoft.com/subscriptions/* using your Microsoft account and follow the instructions provided.

For the latest on Windows Azure

The best way to keep up with new features and enhancements in Windows Azure is by following the official Windows Azure Blog at *http://www.windowsazure.com/en-us/community/blog/*. If you use a newsreader, you can subscribe to the RSS feed for this blog and get the news as it happens.

Ask questions about Windows Azure

There are two great places you can go online to ask questions about Windows Azure and get answers from the community:

- The Windows Azure forums on MSDN at *http://social.msdn.microsoft.com/Forums/windowsazure/en-US/home?category=windowsazureplatform%2Cazuremarketplace%2Cwindowsazureplatformctp*.
- Stack Overflow at *http://stackoverflow.com/questions/tagged/azure*.

About the author

MITCH TULLOCH is a well-known expert on Windows Server administration and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written or contributed to over two dozen books, including the *Windows 7 Resource Kit* (Microsoft Press, 2009), for which he was lead author; *Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter* (Microsoft Press, 2010); and *Introducing Windows Server 2012* (Microsoft Press, 2012), a free e-book that has been downloaded almost three-quarters of a million times.

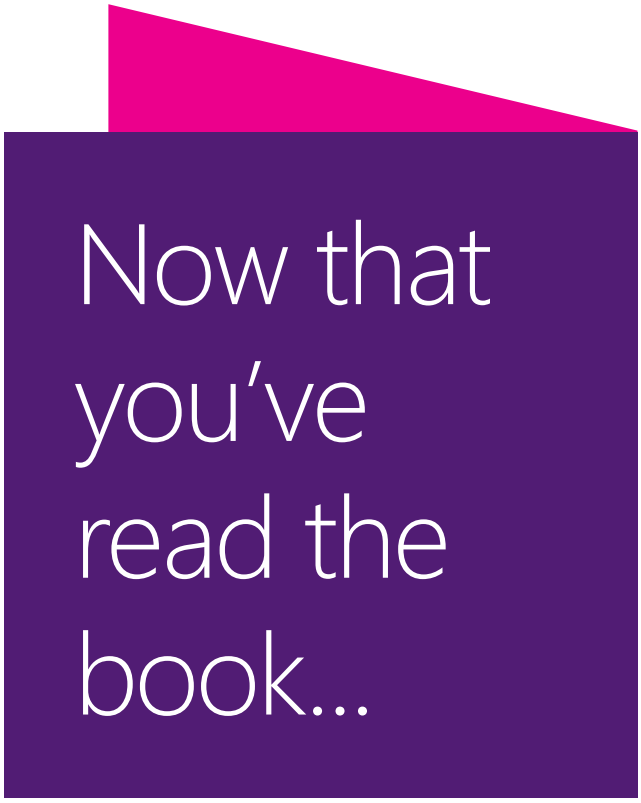
Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is a nine-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at *http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182*.

Mitch is also Senior Editor of WServerNews (*http://www.wservernews.com*), a weekly newsletter focused on system admin and security issues for the Windows Server platform. With more than 100,000 IT Pro subscribers worldwide, WServerNews is the largest Windows Server–focused newsletter in the world.

Mitch runs an IT content development business based in Winnipeg, Canada that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at *http://www.mtit.com*.

You can also follow Mitch on Twitter at *http://twitter.com/mitchtulloch* or like him on Facebook at *http://www.facebook.com/mitchtulloch*



Tell us what you think!

Was it useful?
Did it teach you what you wanted to learn?
Was there room for improvement?

Let us know at http://aka.ms/tellpress

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!

Microsoft