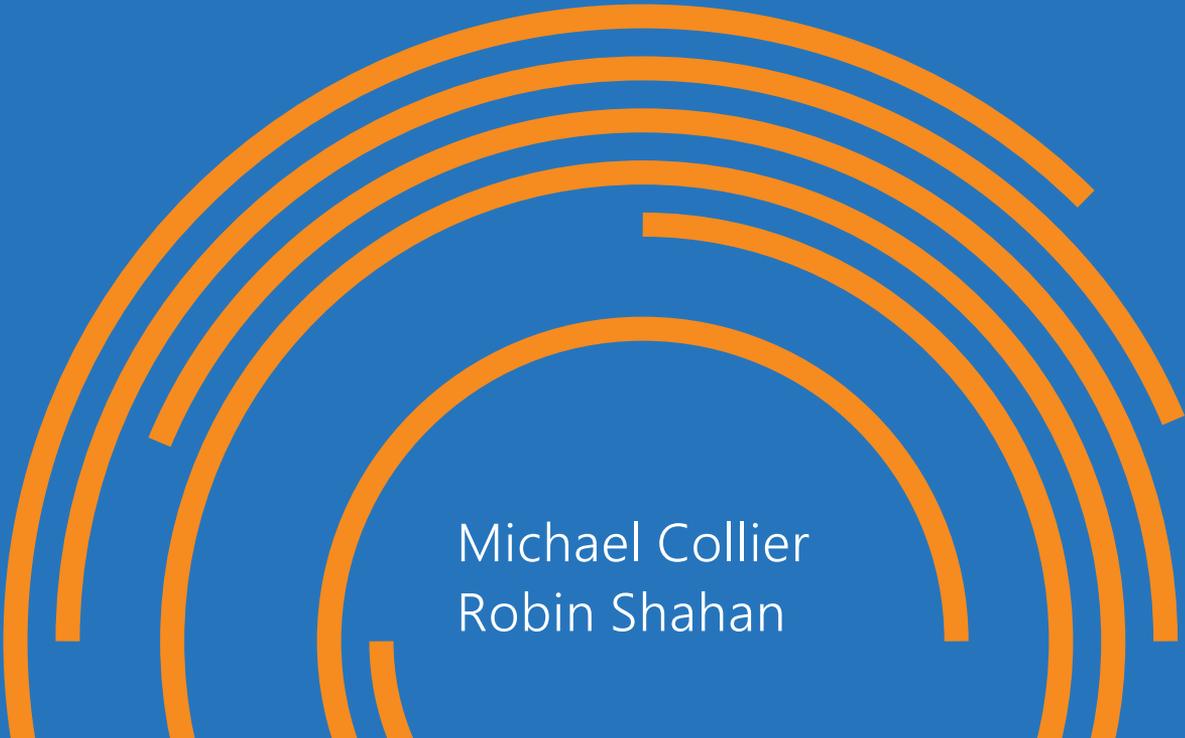


Fundamentals of Azure

Microsoft Azure Essentials



Michael Collier
Robin Shahan

Visit us today at

microsoftpressstore.com

- **Hundreds of titles available** – Books, eBooks, and online resources from industry experts
- **Free U.S. shipping**
- **eBooks in multiple formats** – Read on your computer, tablet, mobile device, or e-reader
- **Print & eBook Best Value Packs**
- **eBook Deal of the Week** – Save up to 60% on featured titles
- **Newsletter and special offers** – Be the first to hear about new releases, specials, and more
- **Register your book** – Get additional benefits



Hear about it first.

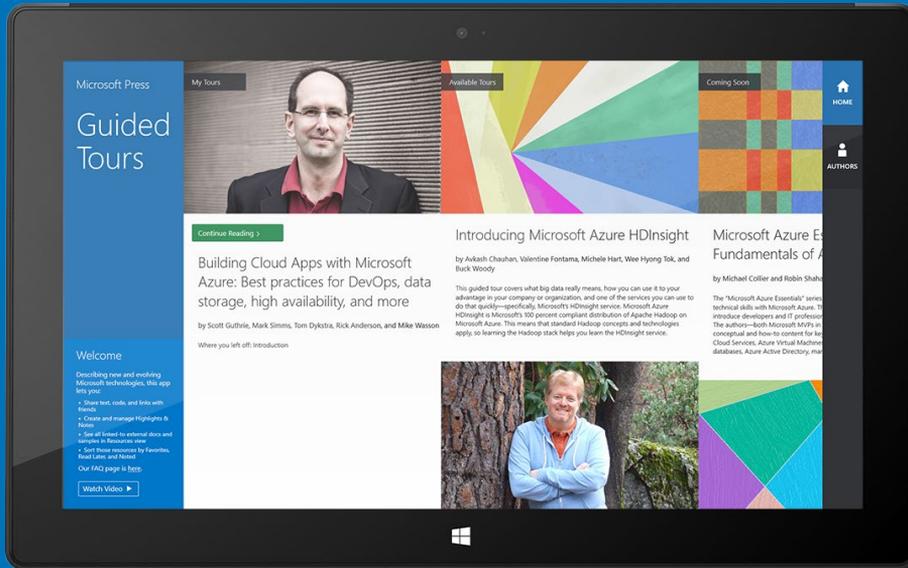


Get the latest news from Microsoft Press sent to your inbox.

- New and upcoming books
- Special offers
- Free eBooks
- How-to articles

Sign up today at MicrosoftPressStore.com/Newsletters

Wait, there's more...



Find more great content and resources in the Microsoft Press Guided Tours app.



The [Microsoft Press Guided Tours](#) app provides insightful tours by Microsoft Press authors of new and evolving Microsoft technologies.

- Share text, code, illustrations, videos, and links with peers and friends
- Create and manage highlights and notes
- View resources and download code samples
- Tag resources as favorites or to read later
- Watch explanatory videos
- Copy complete code listings and scripts





From technical overviews to drilldowns on special topics, get *free* ebooks from Microsoft Press at:

www.microsoftvirtualacademy.com/ebooks

Download your free ebooks in PDF, EPUB, and/or Mobi for Kindle formats.

Look for other great resources at Microsoft Virtual Academy, where you can learn new skills and help advance your career with free Microsoft training delivered by experts.

Microsoft Press

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

Copyright © 2015 Microsoft Corporation. All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-9722-5

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://aka.ms/tellpress>.

This book is provided “as-is” and expresses the authors’ views and opinions. The views, opinions, and information expressed in this book, including URL and other Internet website references, may change without notice.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

Acquisitions, Developmental, and Project Editor: Devon Musgrave

Editorial Production: nSight, Inc.

Copyeditor: Teresa Horton

Cover: Twist Creative

To my wife, Sonja, and son, Aidan; I could not have written this book without your immense support and patience.

—Michael S. Collier

I dedicate this book to my friend Carol Schultz, who passed away earlier this year. She always believed in me.

—Robin E. Shahan

Table of Contents

Foreword	11
Introduction	12
Who should read this ebook.....	12
Assumptions	12
This ebook might not be for you if.....	13
Organization of this ebook.....	13
Conventions and features in this ebook	14
System requirements.....	14
Acknowledgments.....	15
Errata, updates, & support.....	16
Free ebooks from Microsoft Press	16
We want to hear from you	16
Stay in touch.....	16
Chapter 1 Getting started with Azure	17
What is Azure?	17
Overview of cloud computing	17
Comparison of on-premises versus Azure	18
Cloud offering.....	19
SaaS: Software as a Service.....	19
PaaS: Platform as a Service.....	19
IaaS: Infrastructure as a Service	19
Azure services	20
Introduction to portals	20
Azure Preview Portal.....	21
Blades and journeys.....	22

Customizing the Startboard	27
Azure Management Portal	28
Subscription management and billing.....	32
Available subscriptions.....	32
Share administrative privileges for your Azure subscription	33
Add administrative privileges in the Azure Preview Portal.....	34
Add a co-administrator in Azure Management Portal	38
Pricing calculator.....	39
Billing	41
Chapter 2 Azure Websites and Azure Cloud Services.....	46
Creating and configuring websites	46
What is Azure Websites?	46
Creating a new website.....	47
Using the portal.....	47
Websites gallery.....	51
Configure and scale a website	53
Configuration.....	53
Scaling	59
Deploying and monitoring websites	61
Options for creating websites	61
Notepad or an HTML editor	61
WebMatrix	62
Visual Studio	62
Publishing a website from Visual Studio	63
Monitoring a website	65
PaaS Cloud Services.....	66
Creating a cloud service	66

Publishing a cloud service	69
Scaling and monitoring a cloud service.....	72
SCALE options.....	72
Monitoring.....	74
Miscellaneous points.....	77
Configuration in the portal	77
Production and staging slots in the portal	77
Worker roles.....	77
Chapter 3 Azure Virtual Machines.....	78
What is Azure Virtual Machines?.....	78
VM status.....	79
IP address	80
Billing.....	80
Create and configure virtual machines.....	80
Create a virtual machine with the Azure Preview Portal.....	81
Connect to a virtual machine.....	84
Configure disks.....	84
Caching.....	85
Attach a disk.....	85
Formatting disks	87
Disk performance	89
Endpoints.....	89
Virtual machine management.....	91
Availability set.....	92
Service level agreement.....	93
Scalability.....	94
Image capture.....	97

Chapter 4 Azure Storage	101
Blob storage.....	101
Azure Files (preview)	102
Table storage.....	103
Queues.....	104
Redundancy.....	106
Creating and managing storage	107
Create a storage account	107
Create a container and add blobs.....	112
Create a table and add records.....	116
Chapter 5 Azure Virtual Networks.....	119
What is a virtual network	119
Overview.....	119
Definitions	120
Virtual network address spaces.....	120
Subnets.....	120
DNS servers.....	121
Creating a virtual network	121
Quick Create.....	122
Custom Create.....	122
Using a network configuration file.....	125
Export the network configuration file.....	125
Import the network configuration file.....	126
What ifs	127
What if you add a network?	127
What if you remove a network?	127
What if you change a network?	128

Cross-premises connection options.....	129
Site-to-site connectivity.....	129
Point-to-site connectivity	129
Comparing site-to-site and point-to-site connectivity.....	130
Private site-to-site connectivity (ExpressRoute).....	130
Point-to-site network.....	131
Overview of setup process.....	131
Configuring point-to-site VPN	132
Create a virtual network.....	132
Deploy a virtual machine into the virtual network.....	133
Create the VPN gateway.....	136
Create an authentication certificate for your virtual network	137
Upload the self-signed root certificate	140
Install the client certificate (.pfx) to authenticate the virtual network	140
Install the client VPN package	141
Connect to the virtual network through the VPN client.....	141
Connect to the VM using the internal IP address.....	143
Test the VPN connection	145
Chapter 6 Databases	146
Azure SQL Database.....	146
Administration.....	149
Firewall settings	149
Connect using SQL Server Management Studio	153
Billing.....	154
Business continuity	154
Database Copy	158
Import and Export	158

Standard Geo-Replication	162
Active Geo-Replication	162
Geo-Restore	165
Service Level Agreement.....	167
Applications connecting to Azure SQL Database.....	168
SQL Server in Azure Virtual Machines.....	169
Billing.....	169
Virtual machine configuration	170
Business continuity	170
Comparing Azure SQL Database with SQL Server in Azure Virtual Machines.....	171
Database alternatives.....	173
MySQL.....	173
NoSQL options	175
DocumentDB	175
Table storage	175
Chapter 7 Azure Active Directory.....	176
Overview of Azure Active Directory.....	176
What is Azure Active Directory?	176
Active Directory editions.....	177
Creating a directory.....	178
Custom domains.....	180
Delete a directory.....	186
Users and groups.....	186
Add users.....	187
Add groups.....	194
Multi-Factor Authentication	196
Azure Multi-Factor Authentication.....	197

Multi-Factor Authentication for Azure administrators	199
Application gallery	201
Adding gallery applications	201
Assigning users to applications	204
MyApps	206
Chapter 8 Management Tools.....	208
Management tool overview	208
Visual Studio 2013 and the Azure SDK.....	209
Install the Azure SDK.....	210
Manage resources with Server Explorer.....	212
Create an Azure resource.....	218
Windows PowerShell.....	219
Azure PowerShell cmdlet installation.....	220
Connecting to Azure.....	223
Connecting using a Microsoft account	223
Connect using a management certificate	224
Azure PowerShell modes.....	226
Usage	227
Cross-Platform Command-Line Interface.....	228
Installation.....	228
Installing on Windows.....	228
Installing on Linux.....	231
Connecting to Azure.....	232
Connect using a management certificate	232
Connect using a work or school account.....	233
Usage	233
Chapter 9 Business Cases.....	236

Development and test scenarios.....	236
Hybrid scenarios.....	238
Network connectivity.....	238
Internet connectivity.....	239
Application and infrastructure modernization and migration.....	240
Azure Mobile Services.....	241
About the Authors.....	243

Foreword

I'm thrilled to be able to share these Microsoft Azure Essentials ebooks with you. The power that Microsoft Azure gives you is thrilling but not unheard of from Microsoft. Many don't realize that Microsoft has been building and managing datacenters for over 25 years. Today, the company's cloud datacenters provide the core infrastructure and foundational technologies for its 200-plus online services, including Bing, MSN, Office 365, Xbox Live, Skype, OneDrive, and, of course, Microsoft Azure. The infrastructure is comprised of many hundreds of thousands of servers, content distribution networks, edge computing nodes, and fiber optic networks. Azure is built and managed by a team of experts working 24x7x365 to support services for millions of customers' businesses and living and working all over the globe.

Today, Azure is available in 141 countries, including China, and supports 10 languages and 19 currencies, all backed by Microsoft's \$15 billion investment in global datacenter infrastructure. Azure is continuously investing in the latest infrastructure technologies, with a focus on high reliability, operational excellence, cost-effectiveness, environmental sustainability, and a trustworthy online experience for customers and partners worldwide.

Microsoft Azure brings so many services to your fingertips in a reliable, secure, and environmentally sustainable way. You can do immense things with Azure, such as create a single VM with 32TB of storage driving more than 50,000 IOPS or utilize hundreds of thousands of CPU cores to solve your most difficult computational problems.

Perhaps you need to turn workloads on and off, or perhaps your company is growing fast! Some companies have workloads with unpredictable bursting, while others know when they are about to receive an influx of traffic. You pay only for what you use, and Azure is designed to work with common cloud computing patterns.

From Windows to Linux, SQL to NoSQL, Traffic Management to Virtual Networks, Cloud Services to Web Sites and beyond, we have so much to share with you in the coming months and years.

I hope you enjoy this Microsoft Azure Essentials series from Microsoft Press. The first three ebooks cover fundamentals of Azure, Azure Automation, and Azure Machine Learning. And I hope you enjoy living and working with Microsoft Azure as much as we do.

*Scott Guthrie
Executive Vice President
Cloud and Enterprise group, Microsoft Corporation*

Introduction

Microsoft Azure is Microsoft's cloud computing platform, providing a wide variety of services you can use without purchasing and provisioning your own hardware. Azure enables the rapid development of solutions and provides the resources to accomplish tasks that may not be feasible in an on-premises environment. Azure's compute, storage, network, and application services allow you to focus on building great solutions without the need to worry about how the physical infrastructure is assembled.

This ebook covers the fundamentals of Azure you need to start developing solutions right away. It concentrates on the features of the Azure platform that you are most likely to need to know rather than on every feature and service available on the platform. This ebook also provides several walkthroughs you can follow to learn how to create VMs and virtual networks, websites and storage accounts, and so on. In many cases, real-world tips are included to help you get the most out of your Azure experience.

In addition to its coverage of core Azure services, the ebook discusses common tools useful in creating and managing Azure-based solutions. The ebook wraps up by providing details on a few common business scenarios where Azure can provide compelling and valuable solutions.

Who should read this ebook

This ebook focuses on providing essential information about the key services of Azure for developers and IT professionals who are new to cloud computing. Detailed, step-by-step demonstrations are included to help the reader understand how to get started with each of the key services. This material is useful not only for those who have no prior experience with Azure, but also for those who need a refresher and those who may be familiar with one area but not others. Each chapter is standalone; there is no requirement that you perform the hands-on demonstrations from previous chapters to understand any particular chapter.

Assumptions

We expect that you have at least a minimal understanding of virtualized environments and virtual machines. There are no specific skills required overall for this ebook, but having some knowledge of the topic of each chapter will help you gain a deeper understanding. For example, the chapter on virtual networks will make more sense if you have some understanding of networking, and the chapter on databases will be more useful if you understand what a database is and for what you might use one. Web development skills will provide a good background for understanding websites, and some understanding of identity will be helpful when studying the chapter on Active Directory.

This ebook might not be for you if...

This ebook might not be for you if you are looking for an in-depth developer or architecture-focused discussion on a wide range of Azure features, or if you are looking for details on other public or private cloud platforms.

Organization of this ebook

This ebook explores six foundational features of the Microsoft Azure platform, along with insights on getting started with Azure, management tools, and common business scenarios. There are many services in the Azure platform that are not in the scope of this ebook, such as HDInsight (Azure's Hadoop service), Service Bus, and Azure Automation, to mention just a few. To learn about all of the services available in the Azure platform, start your journey at <http://azure.microsoft.com>.

The topics explored in this book include:

- **Getting started with Azure:** Understand what cloud computing is, visit the management portals, and learn about billing.
- **Websites and Cloud Services:** Learn about Azure Websites, from deployment to monitoring, and gain an understanding of the web and worker roles used in Azure Cloud Services.
- **Virtual Machines:** Explore the basic features of Azure Virtual Machines, including how to create, configure, and manage them.
- **Storage:** Read about the basics of Azure Storage, including blobs, tables, queues, and file shares.
- **Virtual Networks:** Learn the basics of virtual networks, including how to create one, and why a virtual network might be necessary. This also covers site-to-site and point-to-site networking, as well as ExpressRoute.
- **Databases:** Explore two relational database options available in Azure: Azure SQL Database and SQL Server in Azure Virtual Machines.
- **Azure Active Directory:** Explore basic features of Azure AD, including creating a directory, users and groups, and using the application gallery.
- **Management Tools:** Explore three common tools for working with Azure: Visual Studio 2013 and the Azure SDK, Azure PowerShell cmdlets, and the Cross-Platform Command-Line Interface
- **Business Scenarios:** Explore four common scenarios for utilizing Azure features: development and test, hybrid, application and infrastructure modernization, and Azure Mobile Services.

Conventions and features in this ebook

This ebook presents information using conventions designed to make the information readable and easy to follow:

- To create specific Azure resources, follow the numbered steps listing each action you must take to complete the exercise.
- There are currently two management portals for Azure: the Azure Management Portal at <http://manage.windowsazure.com> and the new Azure Preview Portal at <http://portal.azure.com>. It is necessary to move between both portals to explore all Azure features; unless otherwise specified, assume the Azure Preview Portal is used.
- Boxed elements with labels such as "Note" or "See Also" provide additional information.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, "Press Alt+Tab" means that you hold down the Alt key while you press Tab.
- A right angle bracket between two or more menu items (e.g., File Browse > Virtual Machines) means that you should select the first menu or menu item, then the next, and so on.

System requirements

For many of the examples in this ebook, you need only internet access and a browser (Internet Explorer 10 or higher) to access the Azure portals.

Chapter 2, "Azure Websites and Azure Cloud Services," and Chapter 4, "Azure Storage," use Visual Studio to show some concepts used in developing applications for Azure. For these examples, you will need Visual Studio. The system requirements are:

- Windows 7 Service Pack 1, Windows 8, Windows 8.1, Windows Server 2008 R2 SP1, Windows Server 2012, or Windows Server 2012 R2
- Computer that has a 1.6GHz or faster processor (2GHz recommended)
- 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
- 20 GB of available hard disk space
- 5400 RPM hard disk drive
- DirectX 9 capable video card running at 1024 x 768 or higher-resolution display

- DVD-ROM drive (if installing Visual Studio from DVD)
- Internet connection

After installing Visual Studio, you must also install the Azure Tools and SDK version 2.4 or higher. You can do this using the Web Platform Installer at <http://azure.microsoft.com/en-us/downloads/>, or, if preferred, you can manually install and configure the required components by following the instructions here: <http://www.microsoft.com/en-us/download/details.aspx?id=43709> (SDK 2.4) or here: <http://www.microsoft.com/en-us/download/details.aspx?id=44938> (SDK 2.5).

Links to both version 2.4 and version 2.5 are provided here because version 2.5 has some breaking changes. If you have any other Azure solutions that use SDK 2.4, you should read the release notes for version 2.5 before upgrading: <http://msdn.microsoft.com/en-us/library/azure/dn873976.aspx>.

The system requirements for the Azure SDK that are not included in the Visual Studio system requirements are as follows:

- IIS7 with ASP.NET and WCF HTTP Activation, Static Content, IIS Management Console, and HTTP Redirection
- Web Deployment Tools 2.1 or up
- Internet Explorer 10 or higher

Depending on your Windows configuration, you might require Local Administrator rights to install or configure Visual Studio 2013.

Acknowledgments

The Azure community is made up of many people bound together by this one technology. We are honored to be members of this community, and we thank you for your help and support. We would like to especially thank Neil Mackenzie, Mike Martin, Gaurav Mantri, and Fabien Lavocat for their detailed technical reviews and feedback. All of them provided additional insights that greatly enhanced the overall quality and value of this ebook.

Special thanks to the team at Microsoft Press for their unwavering support and guidance on this journey. It was a pleasure to work with our editor, Devon Musgrave. Devon provided immensely helpful advice from the days when this ebook was just an idea, all the way through to final copy.

Most importantly, we are profoundly grateful to our families and friends for their love, encouragement, and patience. Many nights and weekends were sacrificed in the writing of this ebook.

Errata, updates, & support

We've made every effort to ensure the accuracy of this ebook. You can access updates to this ebook—in the form of a list of submitted errata and their related corrections—at:

<http://aka.ms/FundAzure>

If you discover an error that is not already listed, please submit it to us at 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 and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to <http://support.microsoft.com>.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

<http://aka.ms/mspressfree>

Check back often to see what is new!

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://aka.ms/tellpress>

We know you're busy, so we've kept it short with just a few questions. Your answers go directly to the editors at Microsoft Press. (No personal information will be requested.) 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

Getting started with Azure

The purpose of this book is to help you understand the fundamentals of Microsoft Azure so you can hit the ground running when you start using it. If you don't have an Azure account, you should consider signing up for a free trial at azure.microsoft.com. If you have an MSDN subscription, you should activate the included Azure benefits and use the associated monthly credit. With an Azure account, you can work through the demos in this book and use them as hands-on labs.

What is Azure?

Overview of cloud computing

With an on-premises datacenter you have to handle and manage everything, including purchasing and installing the hardware, virtualization, installing the operating system and any required applications, setting up the network (including running wires), configuring the firewall, and setting up storage for data. Having done all that, you then become responsible for maintaining it through its entire lifecycle. This imposes both significant capital cost for the hardware and significant operational cost for its maintenance. You do have the luxury of choosing whatever hardware and software you like—but you also have to pay for it, regardless of whether you are using it.

Cloud computing provides a modern alternative to the traditional on-premises datacenter. A public cloud vendor is completely responsible for hardware purchase and maintenance and typically provides a wide variety of platform services that you can use. You lease whatever hardware and software services you require on an as-needed basis, thereby converting what had been a capital expense for hardware purchase into an operational expense. It also allows you to lease access to hardware and software resources that would be too expensive to purchase. Although you are limited to the hardware provided by the cloud vendor, you only have to pay for it when you use it.

Cloud environments typically provide an online portal experience, making it easy for users to manage compute, storage, network, and application resources. For example, a user can use the portal to create a virtual machine (VM) configuration specifying the following: the compute node size (with regard to CPU, RAM, and local disks), the operating system, any predeployed software, the network configuration, and the location of the node. The user then can deploy the VM based on that configuration and within a few minutes access the deployed compute node. This quick deployment compares favorably with the previous mechanism for deploying a VM, which could take weeks just for the procurement cycle.

In addition to the public cloud just described, there are private and hybrid clouds. In a private cloud, you create a cloud environment in your own datacenter and provide self-service access to compute

resources to users in your organization. This offers a simulation of a public cloud to your users, but you remain completely responsible for the purchase and maintenance of the hardware and software services you provide. A hybrid cloud integrates public and private clouds, allowing you to host workloads in the most appropriate location. For example, you could host a high-scale website in the public cloud and link it to a highly secure database hosted in your private cloud (or on-premises datacenter).

Microsoft provides support for public, private, and hybrid clouds. Microsoft Azure, the focus of this book, is a public cloud. The Windows Azure Pack is a free add-on to Microsoft System Center that allows you to host many of the core Azure services in your own datacenter and provide a self-service portal experience to your users. You can integrate these into a hybrid cloud through the use of a virtual private network.

Comparison of on-premises versus Azure

With an on-premises infrastructure, you have complete control over the hardware and software that you deploy. Historically, this has led to hardware procurement decisions focused on scaling up; that is, purchasing a server with more cores to satisfy a performance need. With Azure, you can deploy only the hardware provided by Microsoft. This leads to a focus on scale-out through the deployment of additional compute nodes to satisfy a performance need. Although this has consequences for the design of an appropriate software architecture, there is now ample proof that the scale-out of commodity hardware is significantly more cost effective than scale-up through expensive hardware.

Microsoft has deployed Azure datacenters in 19 regions across the globe from Melbourne to Amsterdam and Sao Paulo to Singapore. Additionally, Microsoft has an arrangement with Via21Net, making Azure available in two regions in China. Only the largest global enterprises are able to deploy datacenters in this manner, so using Azure makes it easy for enterprises of any size to gain the ability to deploy their services close to their customers, wherever they are in the world. And you can do that without ever leaving your office.

For startups, Azure allows you to start with very low cost and scale rapidly as you gain customers. You would not face a large up-front capital investment to create a new VM—or even several new VMs. The use of cloud computing fits well with the scale fast, fail fast model of startup growth.

Azure provides the flexibility to quickly set up development and test configurations. These can be scripted, giving you the ability to spin up a development or test environment, do the testing, and spin it back down. This keeps the cost very low, and maintenance is almost nonexistent.

Another advantage of Azure is that you can try new versions of software without having to upgrade on-premises equipment. For example, if you want to see the ramifications of running your application against Microsoft SQL Server 2014 instead of Microsoft SQL Server 2012, you can create a SQL Server 2014 instance and run a copy of your services against the new database, all without having to allocate hardware and run wires. Or you can run on a VM with Microsoft Windows Server 2012 R2 instead of Microsoft Windows Server 2008 R2.

Cloud offering

Cloud computing usually is classified in three categories: SaaS, PaaS, and IaaS. However, as the cloud matures, the distinction among these is being eroded.

SaaS: Software as a Service

SaaS is software that is centrally hosted and managed for the end customer. It usually is based on a multitenant architecture—a single version of the application is used for all customers. It can be scaled out to multiple instances to ensure the best performance in all locations. SaaS software typically is licensed through a monthly or annual subscription.

Office 365 is a prototypical model of a SaaS offering. Subscribers pay a monthly or annual subscription fee, and they get Exchange as a Service (online and/or desktop Outlook), Storage as a Service (OneDrive), and the rest of the Microsoft Office Suite (online, the desktop version, or both). Subscribers are always provided the most recent version. This essentially allows you to have a Microsoft Exchange server without having to purchase a server and install and support Exchange—the Exchange server is managed for you, including software patches and updates. Compared to installing and upgrading Office every year, this is much less expensive and requires much less effort to keep updated.

Other examples of SaaS include Microsoft One Drive, Dropbox, WordPress, and Amazon Kindle.

PaaS: Platform as a Service

With PaaS, you deploy your application into an application-hosting environment provided by the cloud service vendor. The developer provides the application, and the PaaS vendor provides the ability to deploy and run it. This frees up developers from infrastructure management, allowing them to focus strictly on development.

Azure provides several PaaS compute offerings, including Azure Websites and Azure Cloud Services (web and worker roles). In either case, developers have multiple ways to deploy their application without knowing anything about the nuts and bolts supporting it. Developers don't have to create VMs, use Remote Desktop (RDP) to log into each one, and install the application. They just hit a button (or pretty close to it), and the tools provided by Microsoft provision the VMs and then deploy and install the application on them.

Scaling out an Azure compute service generally is as simple as incrementing the instance count, at which point Azure deploys new VMs and deploys the software on them. Azure even handles the load balancing automatically. To deploy a new version, you just republish, and Azure updates all of the VMs for you.

IaaS: Infrastructure as a Service

An IaaS cloud vendor runs and manages server farms running virtualization software, enabling you to create VMs that run on the vendor's infrastructure. Depending on the vendor, you can create a VM running Windows or Linux and install anything you want on it. Azure also provides the ability to set up

virtual networks, load balancers, and storage and to use many other services that run on its infrastructure. You don't have control over the hardware or virtualization software, but you do have control over most everything else. In fact, unlike PaaS, you are completely responsible for it.

Azure Virtual Machines, the Azure IaaS offering, is a popular choice when migrating services to Azure because it enables the "lift and shift" model for migration. You can configure a VM similar to the infrastructure currently running your services in your datacenter and migrate your software to the new VM. You might need to make tweaks, such as URLs to other services or storage, but many applications can be migrated in this manner.

Azure services

Azure includes many services in its cloud computing platform. Let's talk about a few of them.

- **Compute services** This includes the Microsoft Azure Cloud Services (web and worker roles), Azure Virtual Machines, Azure Websites, and Azure Mobile Services.
- **Data services** This includes Microsoft Azure Storage (comprised of the Blob, Queue, Table, and Azure Files services), Azure SQL Database, and the Redis Cache.
- **Application services** This includes services that you can use to help build and operate your applications, such as the Azure Active Directory, Service Bus for connecting distributed systems, HDInsight for processing big data, the Azure Scheduler, and Azure Media Services.
- **Network services** This includes Azure features such as Virtual Networks, the Azure Content Delivery Network, and the Azure Traffic Manager.

When migrating an application, it is worthwhile to have some understanding of the different services available in Azure, because you might be able to use them to simplify the migration of your application and improve its robustness.

Introduction to portals

An online management portal provides the easiest way to manage the resources you deploy into Azure. You can use this to create virtual networks, use cloud services, set up VMs, set up storage accounts, define websites, and so on, as listed in the previous section.

There currently are two versions of the portal. The one in production is called the Azure Management Portal. The new one under construction is called the Azure Preview Portal.

The two portals have a completely different look and feel, and you navigate each one differently. Not all features (or all parts of all features) have been migrated to the Azure Preview Portal yet. For example, virtual networks are not in the new portal yet; scaling an Azure website has some of the scaling options available in the new portal, but the old portal has all of them. Some features are only in the new portal, such as enabling and managing the Redis Cache.

The current portal is a single resource point of view: you can act on a single resource at a time. The Azure Preview Portal allows you to display and manage multiple resources with a Resource group. In the Preview Portal, you can define a Resource group by specifying a combination of services that work together, such as a website and the database used by that website. This allows you to manage the life cycle of all related assets by managing the Resource group. For more information about Resource groups, check out <http://azure.microsoft.com/en-us/documentation/articles/azure-preview-portal-using-resource-groups/>.

Let's take a look at the two portals and how you navigate through them.

Azure Preview Portal

The Azure Preview Portal is located at *portal.microsoft.com*. When you open this the first time, it will probably look similar to Figure 1-1.

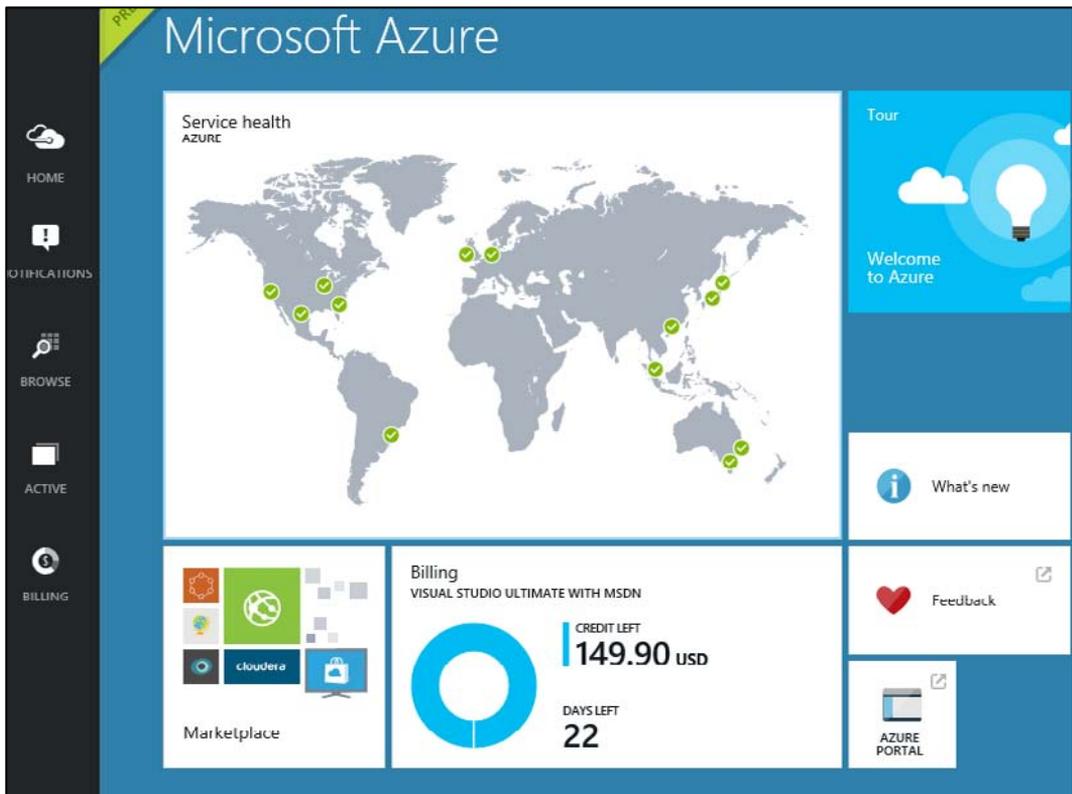


FIGURE 1-1 Azure Preview Portal.

Down the left side, you have buttons for HOME, NOTIFICATIONS, BROWSE, ACTIVE, and BILLING. The central part of the screen with all the tiles is called the Startboard. Figure 1-1 shows the defaults,

which include helpful tiles such as the service health and the billing information for the selected subscription. You can customize your Startboard by accessing the Settings (click your account in the upper-right corner) or right-clicking the Startboard itself.

Let's talk about the buttons on the left side of the screen.

- HOME takes you back to this screen.
- NOTIFICATIONS will display alerts and notifications from the last 24 hours. This includes messages such as "Your VM was created successfully."

BROWSE (Figure 1-2) allows you to filter and jump to some maintenance-related operations. For example, if you administer multiple subscriptions, you can filter so you only see one of them. You can jump to the Portal Settings or the Service Health webpage to see how things are going. Initially, your BROWSE blade will look like the one in Figure 1-2. After you start adding resources to your subscription, more items will show up on the BROWSE blade. Until then, or if the service you're looking for is not displayed, you can browse to Everything by clicking the arrow in the upper-right corner of the screen (Figure 1-3).

- ACTIVE will show the active journeys that you have open. Journeys is covered in more detail in the next section of this chapter.
- BILLING will show your month-to-date billing information. Billing is discussed in the last section of this chapter.

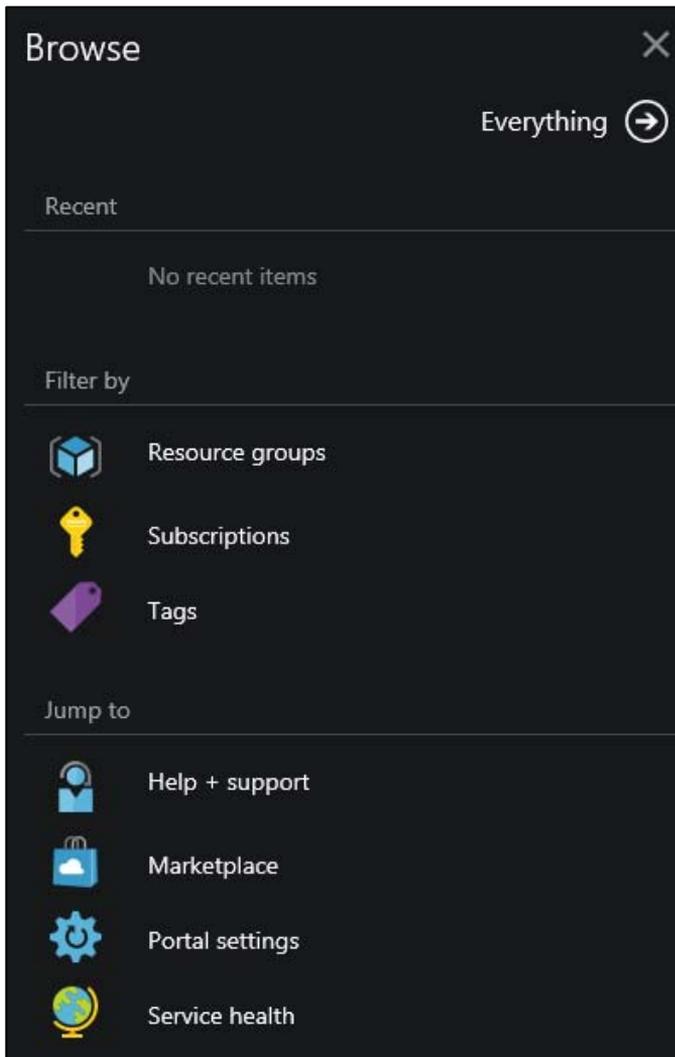


FIGURE 1-2 Browse blade.

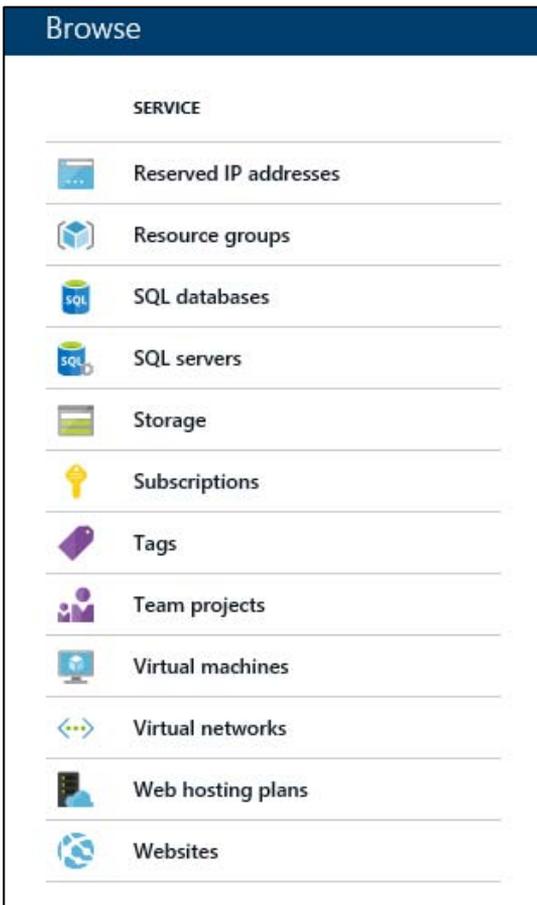


FIGURE 1-3 Browse Everything.

Blades and journeys

The subscription used in this example has three VMs deployed, a virtual network, and a website. Two of the VMs are in the virtual network; the other is not. If we browse to Virtual Machines (BROWSE > Everything > Virtual Machines), we will see three VMs (Figure 1-4). The window to the right that displays the VMs is called a blade.

	VIRTUAL MACHINE	LOCATION	STATUS	SUBSCRIPTION NAME
	ContosoVM1	West US	Running	Visual Studio Ulti...
	contosoVM2	West US	Running	Visual Studio Ulti...
	testazurefiles	West US	Running	Visual Studio Ulti...

FIGURE 1-4 Blade displaying VMs in the subscription.

Next, if we click ContosoVM1, it scrolls to the right and opens another blade showing the properties for ContosoVM1 (Figure 1-5).

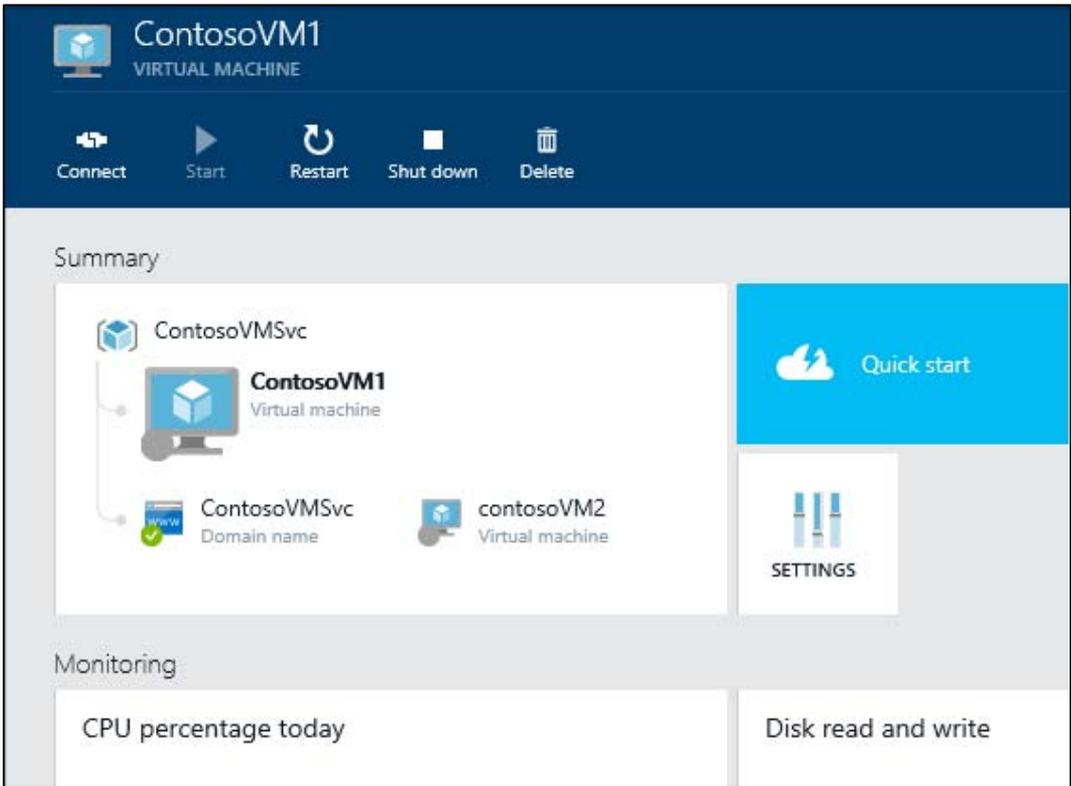


FIGURE 1-5 Blade to the right showing ContosoVM1.

At this point, clicking Settings on this blade opens another blade the right and scrolls to it. This whole chain of selections is called a journey.

At this point, if we go back to the leftmost menu and click BROWSE again, but this time select Virtual Networks, the journey we created to the VM Settings is hidden and it will show a blade for virtual networks instead. Then, if you select a virtual network, it will open a blade to the right showing information for that virtual network. This is another journey.

At this point, clicking ACTIVE will display both of those journeys (Figure 1-6), and you can click either one to go all the way back out to the rightmost blade without re-navigating the entire way.

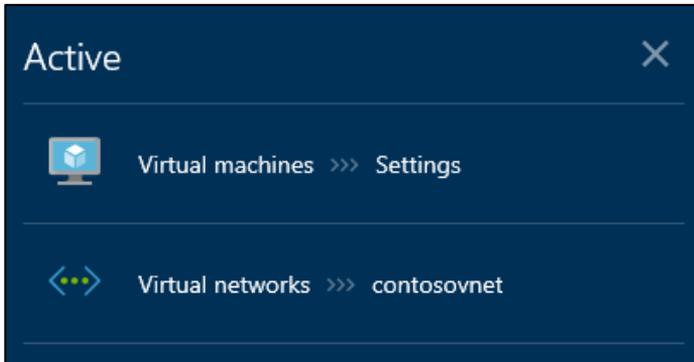


FIGURE 1-6 Window that appears when clicking the ACTIVE button in the left column.

To clear the list of active journeys, you can just hover over one of them and click the X when it appears in the upper-right corner of that row.

Customizing the Startboard

When you create new resources, an Add To Startboard check box is available (Figure 1-7). If you select this check box, it will pin a tile to the Startboard for the new resource that acts as a shortcut to that resource.

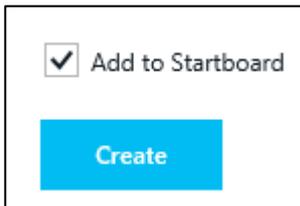


FIGURE 1-7 Add a new resource to the Startboard.

If we create a new VM called testazurefiles and ask for it to be pinned to the Startboard, a new tile shows up as displayed in Figure 1-8.

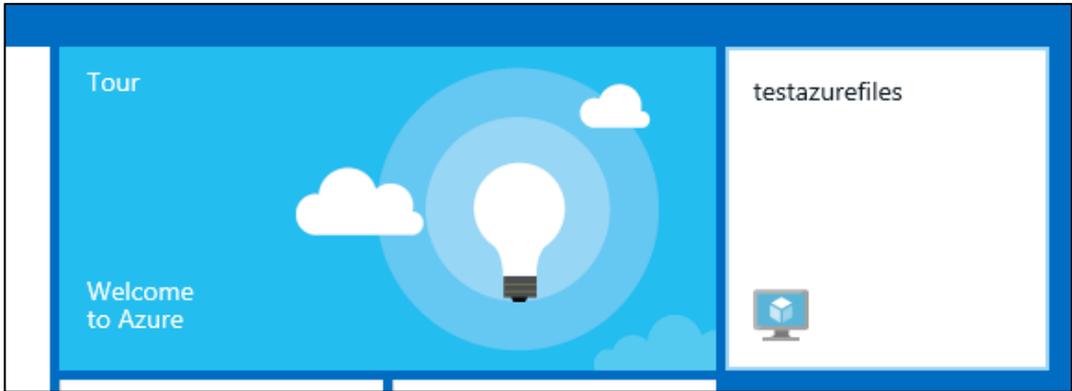


FIGURE 1-8 New resource pinned to the Startboard.

You also can right-click the Startboard to put it in "edit" mode. The options to edit the new tile are shown in Figure 1-9.

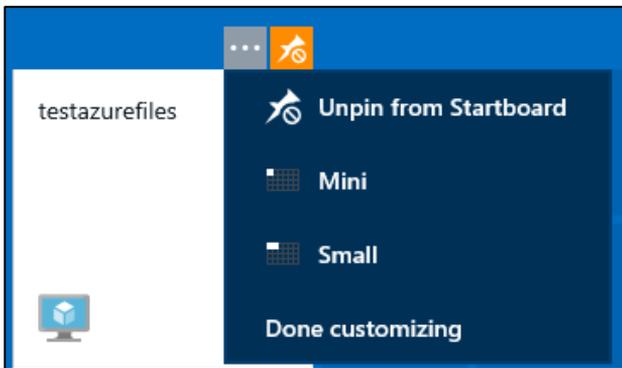


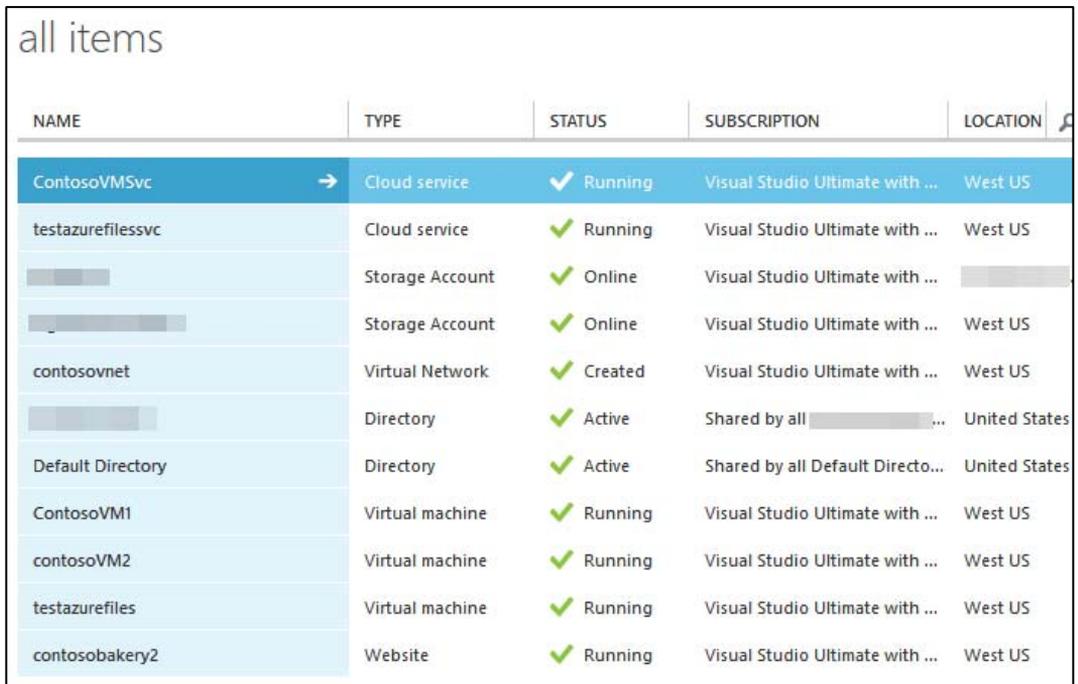
FIGURE 1-9 Customize a new tile.

Be careful when doing this, because if you unpin something you have to track it down and re-pin it. There is no undo option.

Azure Management Portal

The current production version of the Azure Management Portal is at manage.windowsazure.com. Let's navigate to the same subscription used in the sample for the Azure Preview Portal.

After logging in, you can see all of the resources being used in that subscription (Figure 1-10).



The screenshot shows the 'all items' page in the Azure Management Portal. It features a table with columns for NAME, TYPE, STATUS, SUBSCRIPTION, and LOCATION. The table lists various resources such as Cloud services, Storage Accounts, Virtual Networks, Directories, Virtual machines, and Websites, along with their current status and location.

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
ContosoVMSvc	Cloud service	Running	Visual Studio Ultimate with ...	West US
testazurefilessc	Cloud service	Running	Visual Studio Ultimate with ...	West US
	Storage Account	Online	Visual Studio Ultimate with ...	
	Storage Account	Online	Visual Studio Ultimate with ...	West US
contosovnet	Virtual Network	Created	Visual Studio Ultimate with ...	West US
	Directory	Active	Shared by all ...	United States
Default Directory	Directory	Active	Shared by all Default Directo...	United States
ContosoVM1	Virtual machine	Running	Visual Studio Ultimate with ...	West US
contosoVM2	Virtual machine	Running	Visual Studio Ultimate with ...	West US
testazurefiles	Virtual machine	Running	Visual Studio Ultimate with ...	West US
contosobakery2	Website	Running	Visual Studio Ultimate with ...	West US

FIGURE 1-10 All items in the Azure Management Portal.

On the left side of the screen is a list of resource types (Figure 1-11). The displayed list is abbreviated for space; it actually includes all resource types.



FIGURE 1-11 Left side menu listing all resource types.

This list allows you to look at a specific resource type. For example, to look at your VMs, click VIRTUAL MACHINES. In our case, this brings up the list of VMs (Figure 1-12).

The image shows the "virtual machines" page in the Azure portal. It has tabs for "INSTANCES", "IMAGES", and "DISKS". Below the tabs is a table with columns: NAME, STATUS, SUBSCRIPTION, LOCATION, and DNS NAME. The table contains three rows of VM instances.

NAME	STATUS	SUBSCRIPTION	LOCATION	DNS NAME
ContosoVM1	Running	Visual Studio Ultimate ...	West US	contosoovmsvc.clo
contosoVM2	Running	Visual Studio Ultimate ...	West US	contosoovmsvc.clo
testazurefiles	Running	Visual Studio Ultimate ...	West US	testazurefilessvc.cl

FIGURE 1-12 The list of the resources that are virtual machines.

Click ContosoVM1. This opens the Dashboard for that VM (Figure 1-13).



FIGURE 1-13 VM Dashboard.

The menu for navigating the various features of the VM is across the top: DASHBOARD, MONITOR, ENDPOINTS, and CONFIGURE. There is a summary for the performance of the VM and various general information such as the DNS NAME available.

The menu options and Dashboard change depending on the type of resource, and the contents of the menu options also rely on the type of resource. For example, the CONFIGURE screen for a website has a .NET version, PHP version, Java version, and Python version, but these are not relevant to a VM, so they don't show up on a VM's Dashboard.

Across the bottom is a menu, or command bar, that is context aware. It changes based on the resource being acted on. For example, for VMs it looks like Figure 1-14.



FIGURE 1-14 Actions for the selected resource, which in this case is a VM.

This menu also changes depending on the displayed resource.

Notifications are displayed at the bottom of the screen just above the command bar (Figure 1-15).

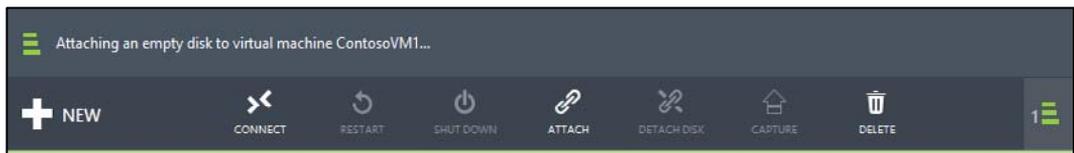


FIGURE 1-15 Notification messages.

Subscription management and billing

Available subscriptions

There are several different kinds of subscriptions providing access to Azure services. You must have a Microsoft account (created by you for personal use) or a work or school account (issued by an administrator for business or academic use) to access these subscriptions.

Let's take a look at the most common subscriptions:

- **Free trial** The link to sign up for a free trial is on the front page of azure.com. This gives you a \$200 credit and a month to try out any combination of resources in Azure. If you exceed your credit amount, your trial will be suspended. At the end of the trial, your services will be decommissioned and will no longer work. You can upgrade this to a pay-as-you-go subscription at any time.
- **MSDN subscriptions** If you have an MSDN subscription, you get a specific amount in Azure

credit each month. For example, if you have a Visual Studio Premium with MSDN subscription, you get \$100 per month in Azure credit.

If you exceed the credit amount, your service will be disabled until the next month starts. You can turn off the spending limit and add a credit card to be used for the additional costs. Some of these costs are discounted for MSDN accounts. For example, you get a 33 percent discount on Windows Virtual Machines. This makes MSDN accounts ideal for development and test scenarios.

For more information and to see the available MSDN subscription tiers, check out <http://azure.microsoft.com/en-us/pricing/member-offers/msdn-benefits-details/>. Note that these subscriptions are to be used for development and testing, not for production.

- **BizSpark accounts** The BizSpark program provides a lot of benefits to startups, not the least of which is access to all of Microsoft's software for development and test environments for up to five MSDN accounts. In addition to these benefits, you get \$150 in Azure credit for each of those five MSDN accounts, and you pay reduced rates for several of the Azure services, such as Windows Virtual Machines.

For more information, check out <http://azure.microsoft.com/en-us/offers/ms-azr-0064p/>.

- **Pay-as-you-go** With this subscription, you pay for what you use by attaching a credit card or debit card to the account. If you are an organization, you also can be approved for invoicing.

For more information, check out <http://azure.microsoft.com/en-us/offers/ms-azr-0003p/>.

- **Enterprise agreements** With an enterprise agreement, you commit to using a certain amount of services in Azure over the next year, and you pay that amount ahead of time. The commitment that you make is consumed throughout the year. If you exceed the commitment amount, you can pay the additional usage quarterly or annually. Depending on the amount of the commitment, you get a discount on the services in Azure.

For more information, check out <http://azure.microsoft.com/en-us/pricing/enterprise-agreement/>.

Share administrative privileges for your Azure subscription

Once you have signed up for an Azure subscription with your Microsoft account, you can give administrative access to additional Microsoft accounts. This is done differently depending on whether you are using the Azure Management Portal or the Azure Preview Portal. If you want the new account to be able to access both portals, you need to give it access in both portals.

This is because the Azure Preview Portal uses Role-Based Access Control (RBAC) and the Azure Management Portal does not. RBAC is a feature that allows you to grant more granular permissions to account management than just full access to a subscription. (For more information about RBAC, see <http://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-configure/>.)

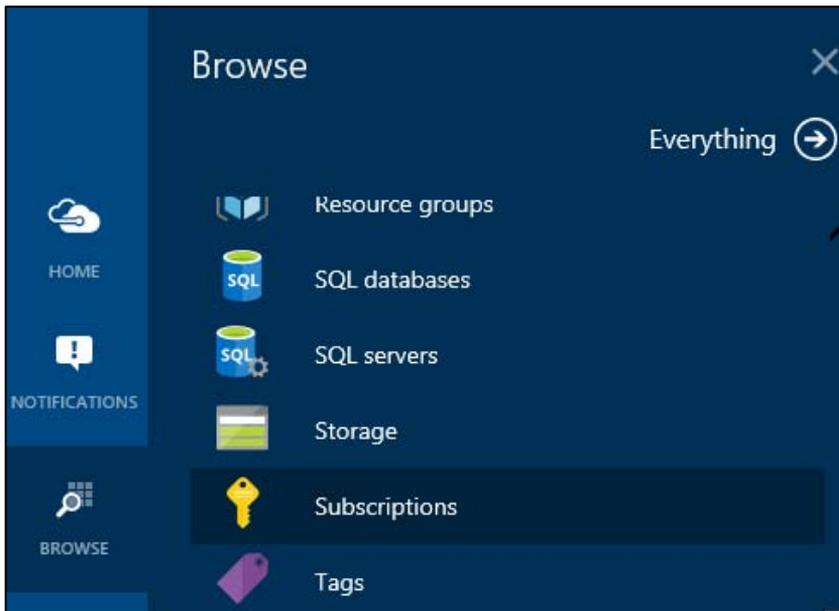
- To give someone access to modify the resources in a subscription in the Azure Preview Portal, you add the OWNER role for the subscription to the user's account.
- To grant administrative access to an account in the Azure Management Portal, you add the user's account as a co-administrator to the subscription. This account will have all of the same privileges as the owner of the original subscription, but it does not allow the user to change the service administrator or add and remove other co-administrators.

In both portals, when requesting that an account be given access, if the account is not already in the subscription's default Azure Active Directory it is added automatically. You can view and administer the users in the Active Directory in the Azure Management Portal.

Let's look at how we can give someone access to a subscription in each of the portals.

Add administrative privileges in the Azure Preview Portal

1. Go to the Azure Preview Portal (portal.azure.com) and log into your Azure account.
2. Click BROWSE on the left side of the screen and then select Subscriptions.



3. On the Subscriptions blade that opens, you should see at least one subscription. Click the one to which you want to add an administrator. This opens the Subscription blade.

Visual Studio Ultimate with MSDN

SUBSCRIPTION

Overview

Visual Studio Ultimate with MSDN
MONETARY CREDIT



CREDIT LEFT
119.70 USD

DAYS LEFT
8



Visual Studio
Ultimate with...



Visual Studio
Ultimate with...

Access

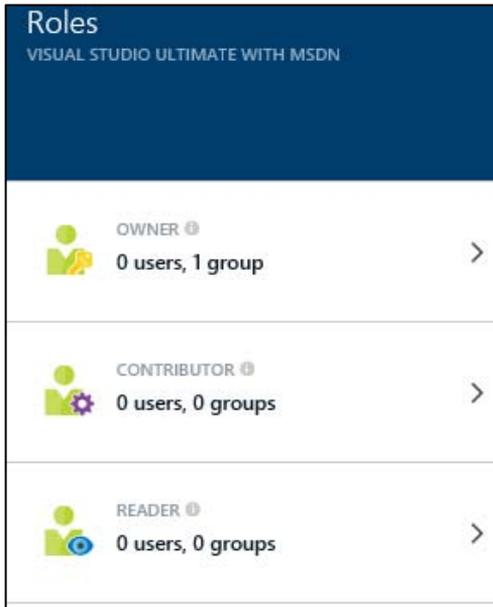
Roles

Owner	0	1
Contributor	0	0
Reader	0	0

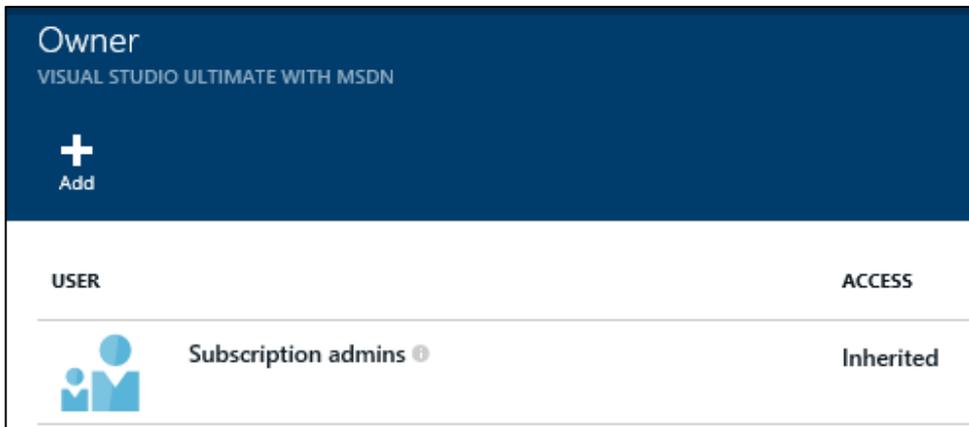
Users

0 users **1** group

4. In the Access section, click the Roles tile to open the list of roles.

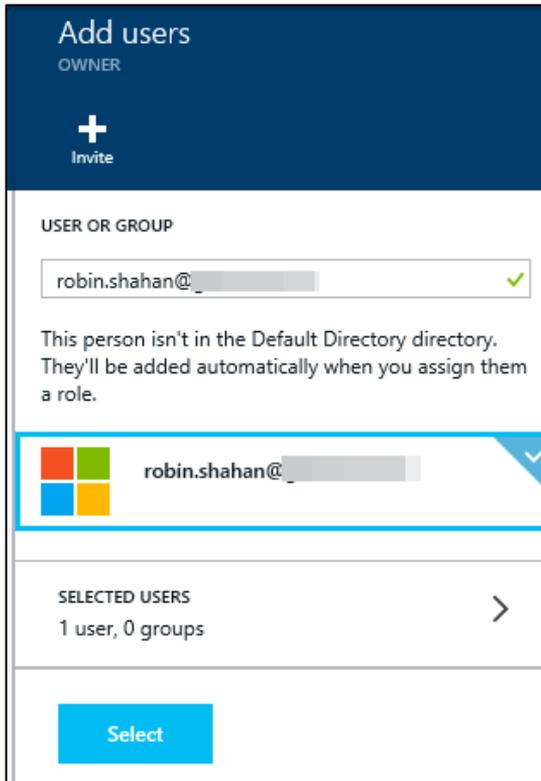


- To add an administrator, you add the account to the OWNER role. Click OWNER to open the Owner blade.



In this account, there is only one user with the Owner role: Subscription admins. If you click the owner, you can see the role, the group to which it is assigned (Subscription admins), and the subscription to which it is assigned (Visual Studio Ultimate with MSDN, in this case).

- Click +Add at the top of the Owner blade. This opens the Add Users blade, where you specify the account to which you want to grant OWNER access. You can select an account from the displayed entries from the default Active Directory or search for an account by name or email address. If the account is not in the default Active Directory, you will see a message that it will be added automatically.



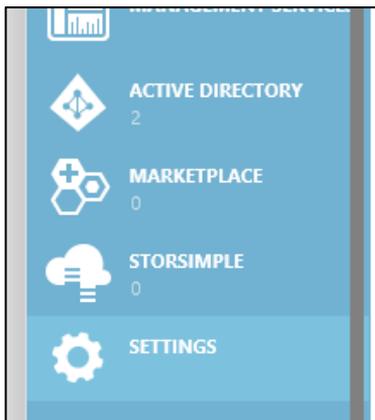
7. Select the account and then click Select at the bottom of the blade. It will refresh the Owner blade, and now you can see that the account has been assigned to the Owner role. You can log in with that account and administer the subscription.

Owner	
VISUAL STUDIO ULTIMATE WITH MSDN	
+ Add	
USER	ACCESS
 robin.shahan@ [redacted]	Assigned
 Subscription admins ⓘ	Inherited

This account now has administrative privileges to the Azure Preview Portal, but not to the other portal.

Add a co-administrator in Azure Management Portal

1. Log into the Azure Management Portal (manage.windowsazure.com). Click SETTINGS on the left side.



2. On the SETTINGS screen, select ADMINISTRATORS and then click ADD+ at the bottom of the screen to open a screen where you can specify the co-administrator.

ADD A CO-ADMINISTRATOR

Specify a co-administrator for subscriptions

Co-administrators can fully manage the services within a subscription. Enter a valid email address, and then select at least one subscription.

EMAIL ADDRESS

robin.shahan@ [redacted]   Microsoft Account

SUBSCRIPTION	SUBSCRIPTION ID
<input checked="" type="checkbox"/> Visual Studio Ultimate with MSDN	[redacted]

3. Type the email address and select the subscription you want that person to be able to administer and then select the check box in the lower-right corner of the screen. It will add this to the ADMINISTRATORS screen and show the subscription for which the account is now a co-administrator.

settings

SUBSCRIPTIONS MANAGEMENT CERTIFICATES ADMINISTRATORS AFFINITY GROUPS USAGE

REMOTEAPP **PREVIEW**

NAME	SUBSCRIPTION	SUBSCRIPTION ID	ROLE
me@robinshahan.com	Visual Studio Ultimate with MS...	[redacted]	Service administrator
robin.shahan@ [redacted]	Visual Studio Ultimate with MS...	[redacted]	Co-administrator

This account can now administer the subscription in the Azure Management Portal.

Pricing calculator

Pricing for your Azure infrastructure can be estimated using the pricing calculator found at <http://azure.microsoft.com/en-us/pricing/calculator/>.

The pricing for each service in Azure is different. Many Azure services provide basic, standard, and premium tiers, usually with several price and performance levels in each tier, allowing you to select an appropriate performance level for your use of the service. As you change the selections, the pricing estimate is provided at the bottom of the page. You can look at each feature separately or select the full calculator to estimate multiple features together.

For example, click Websites to view the Website pricing (Figure 1-16). Click Standard and set the size and number of VMs and then select a value for Bandwidth (ingress is free). The price is calculated and totaled at the bottom.

Websites

Free Shared Basic **Standard**

Load balance unlimited websites per region in up to 10 auto-scaled dedicated instances. Includes support for custom domains and SSL, built-in FTP, Git, TFS and Web Deploy support.

3 VMs **\$223.20**
\$0.30/hr

S M L
Small VM (1.6 GHz CPU, 1.75 GB RAM)

Bandwidth

103GB **\$11.76**

United States + Europe egress

Support

Included **\$0.00**

\$234.96/mo

Currency:
US Dollar (\$)

Estimated Price

FIGURE 1-16 Pricing calculator with Websites selected.

Each feature has its own selections that you can specify. For example, if you select Virtual Machines, you can specify the following:

- Which type of VMs you want and the count for each selected:

- Windows VM: Basic or Standard, D-Series or A-Series
- Linux VM: Basic or Standard, D-Series or A-Series
- SQL Server VM: Basic or Standard; D-Series or A-Series; Web, Standard, or Enterprise edition
- Biztalk Server: Basic or Standard, Standard or Enterprise edition
- Oracle: Basic or Standard, multiple selections of Oracle software
- Bandwidth (egress)

In another example of the selections being service-dependent, if you select Mobile Services, you can specify the following:

- Tier (Free, Basic, Standard)
- Whether to include a SQL Database
- Bandwidth (egress)
- The number of push notifications per month

The pricing calculator can be very helpful in estimating your Azure costs. Note that it does not include variations by region, but you can find those if you go to the individual service pricing pages at <http://azure.microsoft.com/en-us/pricing/> and select the region.

Billing

An important component of using Azure is being able to view your billing information. If you have an account such as an MSDN account that allows you a certain amount of credit, it's nice to know how much you have left and to view where the costs are accumulating. This shows up by default in the Startboard of the Azure Preview Portal (Figure 1-17).

Microsoft Azure

Service health
AZURE



Marketplace

Billing
VISUAL STUDIO ULTIMATE WITH MSDN



CREDIT LEFT
119.65 USD

DAYS LEFT
7

FIGURE 1-17 Billing tile on the Azure Preview Portal Startboard.

This indicates that this account has \$119.65 in credit left, and it is seven days until it starts a new cycle and the full credit is applied again.

You also can click the BILLING selection in the menu on the left side of the Startboard to see this information. If you click that tile or select the subscription on the BILLING blade, you can get more information (Figure 1-18).

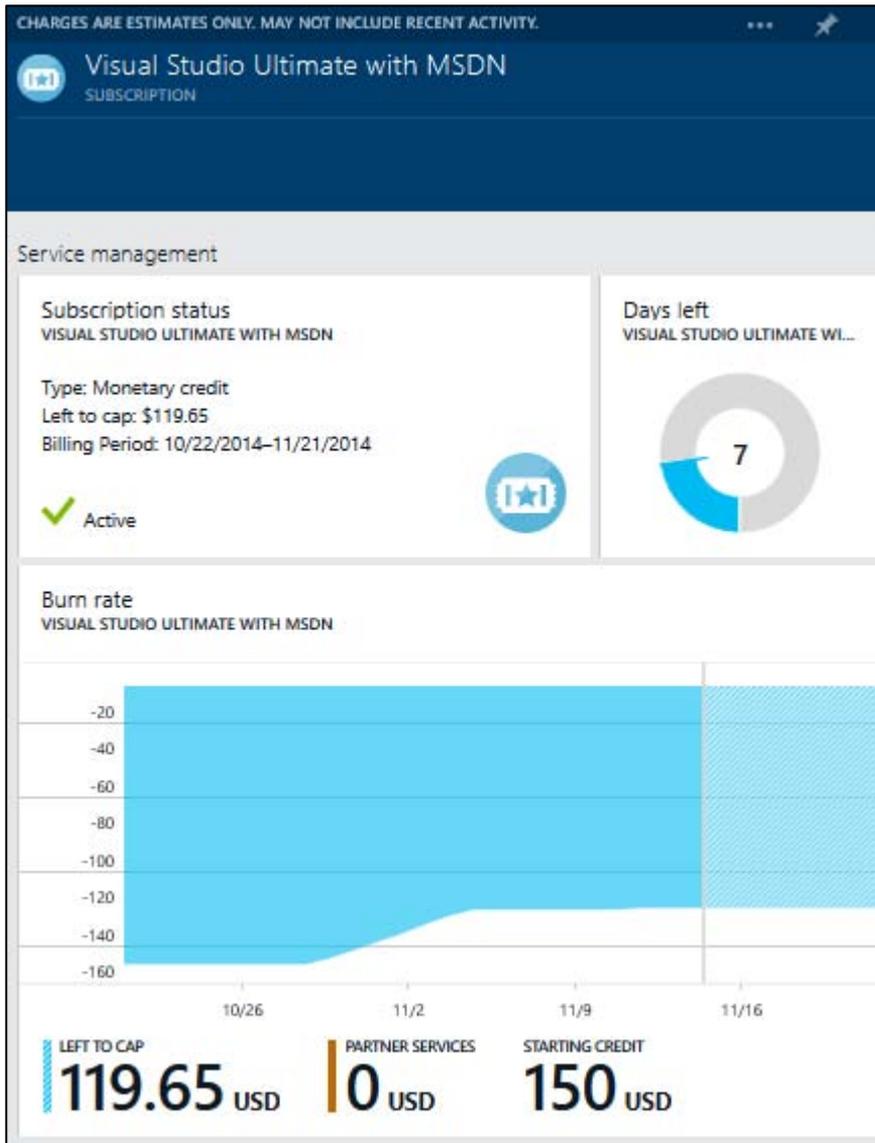


FIGURE 1-18 Billing blade.

This shows the amount of credit left, the billing period information, and the burn rate (the rate at which the cost is accumulating). At the bottom of that blade, it shows the breakdown of charges by Azure service (Figure 1-19) so you can see where the costs are.

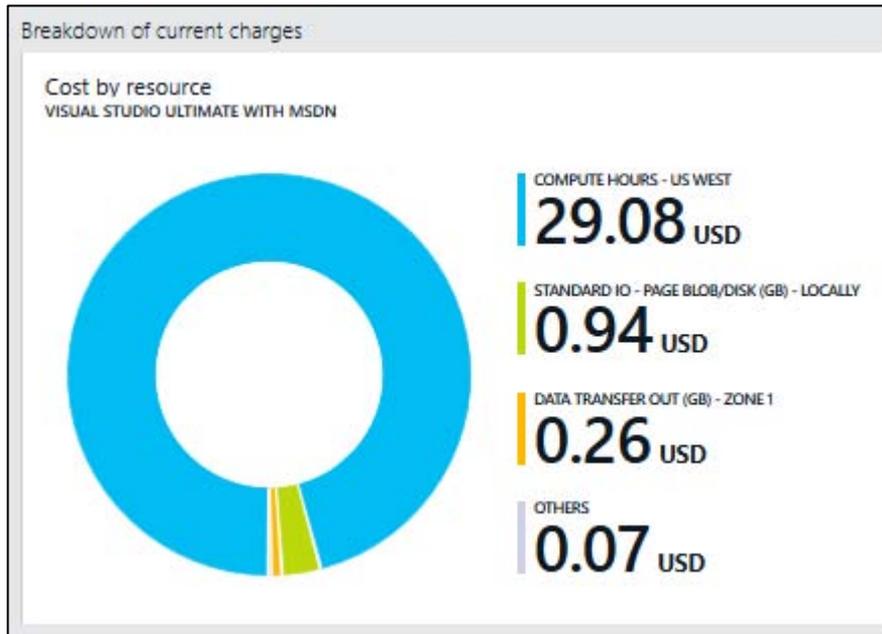


FIGURE 1-19 Breakdown of charges.

If you click this tile, it will show you the details of the Resource Costs (Figure 1-20).



NAME	RESOURCE GUID	CONSUMED UNITS	BILLABLE UNI...	TOTAL (USD)
Compute Hours - US West	a3ab71e8-069...	484.68	484.68	29.08
Standard IO - Page Blob/Di...	d23a5753-ff85...	18.73	18.73	0.94
Data Transfer Out (GB) - Zo...	9995d93a-7d3...	3.02	3.02	0.26
Storage Transactions (in 10...	964c283a-83a...	205.35	205.35	0.07
Data Transfer In (GB) - Zon...	32c3ebec-164...	0.04	0.04	0.00
Free Websites - Websites	c0f5cb45-6fb1...	0.50	0.50	0.00
Standard IO - Block Blob (...	0f521674-5eb...	0.06	0.06	0.00

FIGURE 1-20 Billing details.

The ability to view the billing information on a regular basis is very helpful when managing the costs for your Azure subscription. If you have a subscription with a monthly credit, you can tell when you're getting close to the cap. You also can tell where your costs are accumulating. Also, if you provision some VMs and forget they're out there, you'll be able to see them because they will have billing associated with them.

Chapter 2

Azure Websites and Azure Cloud Services

In this chapter, we take a look at two of the Platform as a Service (PaaS) offerings in Microsoft Azure: Azure Websites and Azure Cloud Services. We talk about what Azure Websites is, how to use the service to create websites, and how to keep them updated. We also look at the options for prebuilt websites offered by Azure.

Cloud Services is a PaaS compute feature in which applications are deployed into instances (virtual machines [VMs]) of server types referred to as web roles and worker roles. The deployment of the instances is fully managed by Microsoft, making it easy to scale applications in and out. This feature is not to be confused with the Cloud Services that are a container for self-deployed VMs, which are covered in Chapter 3, “Azure Virtual Machines.”

Creating and configuring websites

In this section, we take a look at what Azure Websites is, discuss some of its features, and show you how to create, configure, and scale websites.

What is Azure Websites?

Azure Websites is a managed cloud service that allows you to deploy a web application and make it available to your customers on the Internet in a very short amount of time. You don't directly support the VMs on which your website runs; they are managed for you.

Supported languages include .NET, Java, PHP, Node.js, and Python. In addition to creating your own website, there are several web applications available to use as a starting point, such as WordPress, Umbraco, Joomla!, and Drupal.

You can use continuous deployment with Team Foundation Server (TFS), Git, or GitHub so that every time you commit a change, a new version of the website is deployed.

You have the ability to scale the number of instances in and out on demand; you also can configure autoscaling so Azure will scale it in or out for you depending on specific performance measures such as CPU Percentage. If your website has multiple instances, you can configure load balancing to make the most of your resources.

For diagnostics, you can gather performance statistics, application logging, web server logging, IIS

logs, and IIS Failed Request logs. If you're using Microsoft Visual Studio, you can even remotely debug your application while it is running in the cloud.

In short, there are many features available in Azure Websites to make it easy for you to deploy, manage, and troubleshoot a web application.

Creating a new website

Let's create a new website. Later in this section, we will publish content to the website.

Start by logging into the Microsoft Azure Preview Portal (portal.azure.com). At this point, you need an Azure account. If you don't have one, you can sign up for a free trial at azure.microsoft.com.

Using the portal

After logging into the portal, click the big +NEW icon in the lower-left corner of the screen and select Website, as displayed in Figure 2-1.

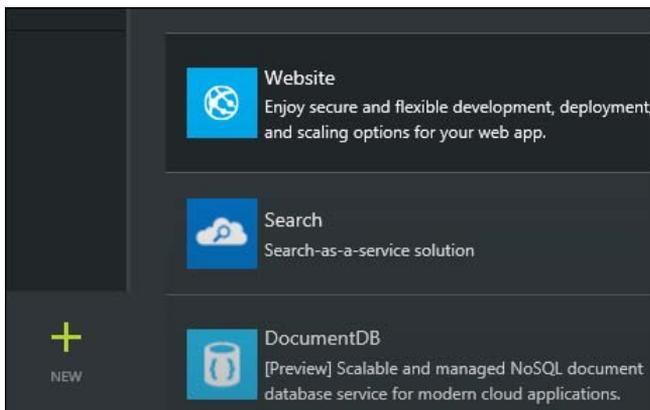
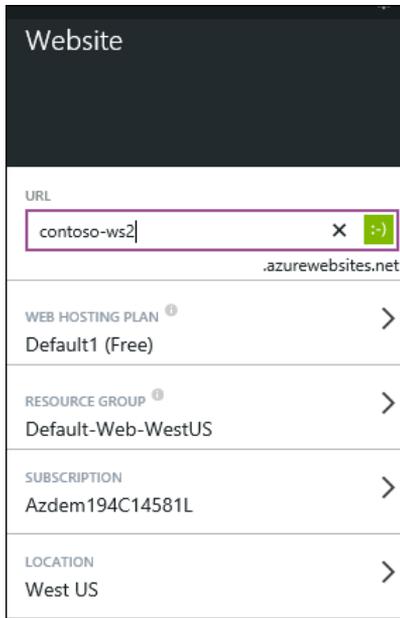


FIGURE 2-1 Add a new website in the Azure Preview Portal.

You should now see something similar to Figure 2-2, with the fields ready to be filled in.



The screenshot shows a 'Website' creation form with the following fields:

- URL:** A text input field containing 'contoso-ws2', followed by a green square with a smiley face icon and the text '.azurewebsites.net'.
- WEB HOSTING PLAN:** A dropdown menu with 'Default1 (Free)' selected and a right-pointing chevron.
- RESOURCE GROUP:** A dropdown menu with 'Default-Web-WestUS' selected and a right-pointing chevron.
- SUBSCRIPTION:** A dropdown menu with 'Azdem194C14581L' selected and a right-pointing chevron.
- LOCATION:** A dropdown menu with 'West US' selected and a right-pointing chevron.

FIGURE 2-2 Create a new website.

The URL must be unique among all of the entries used in Azure Websites. If accepted, there will be a green square with a smiley face in it. Note that whatever prefix is provided here will be appended with *.azurewebsites.net* to create the URL for the website.

SUBSCRIPTION shows the name of the subscription assigned to the Microsoft account with which you logged in. If you administer multiple accounts with the same Microsoft account, you can click SUBSCRIPTION and select the subscription you want to use. LOCATION is the region of the datacenter where the website will be hosted. Select the LOCATION closest to you. Accept the default for RESOURCE GROUP.

WEB HOSTING PLAN defines the allocation of resources for the website, such as number of cores and memory, amount of local storage, and the features available, such as autoscaling and backups.

If you click the selection for WEB HOSTING PLAN, the window in Figure 2-3 will be displayed. You can specify a name for a new web hosting plan and then pick the plan you want. Not all of the plans are displayed on that screen. If you scroll down below the OK button (past what is displayed here), you will see a BROWSE ALL PRICING TIERS option, and you can click that to see all of them. Under that selection is Or Use Existing, which basically says not to create a new web hosting plan. Select the free tier or, if your default is the free tier, select Use Existing.

Web hosting plan

Choose a new or existing web hosting plan. [Learn more](#)

 Create new

NAME

S1 STANDARD 74.40	B1 BASIC 55.80	S3 STANDARD 297.60
--------------------------------	-----------------------------	---------------------------------

1 Core
1.75 GB RAM
 Storage 50 GB
 Custom domains / SSL 5 SNI, 1 IP
 Auto scale Up to 10 instances
 Backup Daily
 Website staging 5 slots
 Geo availability Traffic Manager

FIGURE 2-3 Web Hosting Plan selection.

Use the defaults on the rest of the fields, make sure the Add To Startboard check box is selected, and click Create on the bottom of the new website screen (Figure 2-4).

Add to Startboard

FIGURE 2-4 Create a website and add it to the Startboard.

Azure will create your new website, pin it to the Startboard of your portal so you can easily find it, and show the website and its properties, as displayed in Figure 2-5.

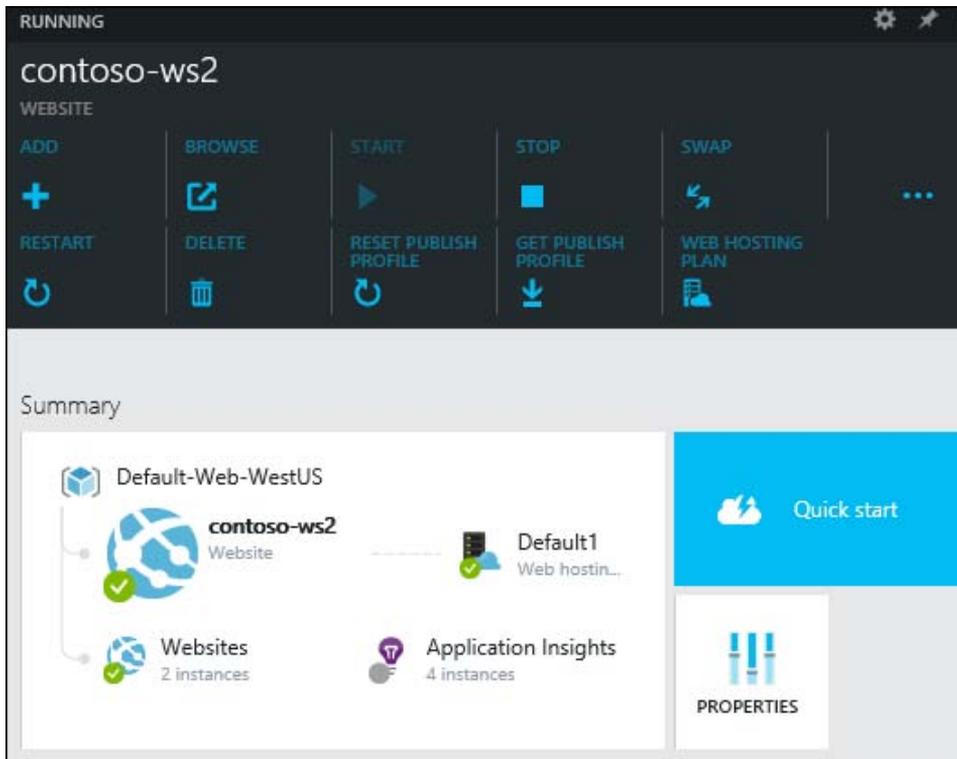


FIGURE 2-5 Website options.

If you click the three dots to the right of SWAP, you can see all of the options:

- ADD adds a new website.
- BROWSE opens your website in the browser. If you haven't published anything yet, it shows a default page directing you to various deployment tools.
- START/STOP starts and stops the website.
- SWAP swaps deployment environments. For example, if you have a production environment and a staging environment, you can publish your website to staging and test it. When you're satisfied with it, you can promote it to production by using the SWAP option and then remove the staging environment that is now the old production version.
- RESTART restarts your website.
- DELETE removes the website from your account.
- RESET PUBLISH PROFILE resets the publishing credentials and invalidates the old credentials;

these are the credentials used for FTP and Git access.

- GET PUBLISH PROFILE retrieves the information needed to publish a website from Visual Studio.
- WEB HOSTING PLAN allows you to change the size, instance count, etc. for the host on which the website is running.

At this point, you've created a new Azure website but you haven't published any content to it; we'll do that in the section "Publishing a website from Visual Studio" later in this chapter.

Websites gallery

While we're looking at the Azure Preview Portal, let's look at some of the website options available from the Azure Marketplace. Click BROWSE on the left side of the page and then select Marketplace. In the Marketplace blade, select Web to display the options for websites.

Here, there are several precreated websites and templates that you can take advantage of. If you scroll down, you can see the categories. At the end of any row, clicking More will show additional options in that category. Here are some of the choices displayed:

- Blogs + CMSs: WordPress, DNN, Joomla!, Umbraco CMS, MonoX, and Drupal
- Starter Sites: ASP.NET, HTML5, Node.js, PHP, some examples like a Bakery website

Select Scalable WordPress—it shows you details on the right side. Click Create at the bottom of that window. This opens a window where you can configure your WordPress site; see Figure 2-6.

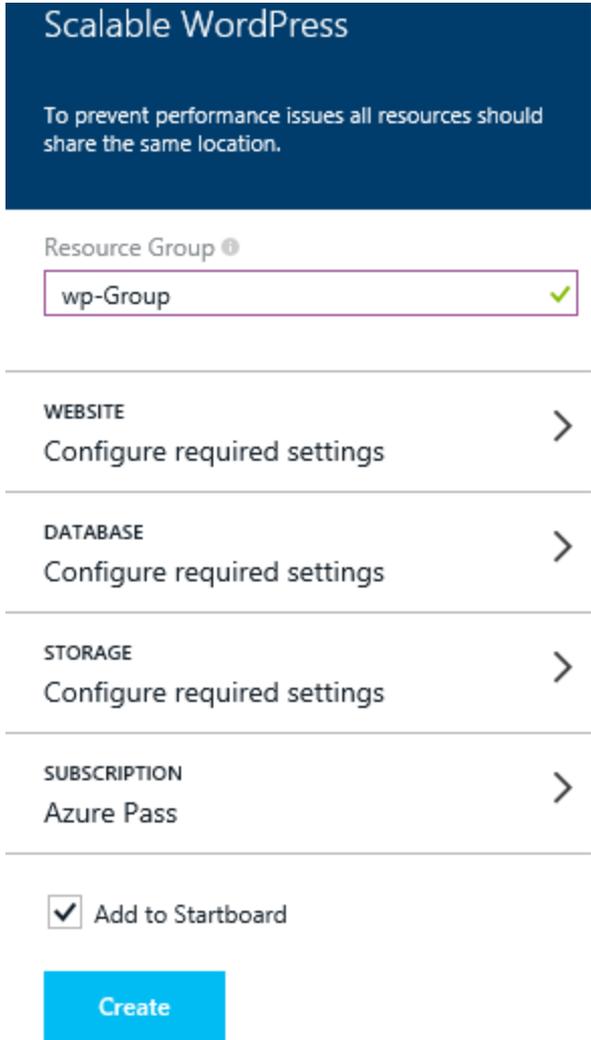


FIGURE 2-6 Set up and configure your WordPress website.

RESOURCE GROUP is a way of grouping multiple resources to be used to see and manage Azure resources that are related to one another, such as a website and a database. Fill that in and under WEBSITE, click Configure Required Settings.

WEBSITE settings include the WEB HOSTING PLAN (the same as discussed previously), the LOCATION (the region in which the datacenter resides), and the WEB APP SETTINGS (including various keys and SALT values).

DATABASE settings include the DATABASE NAME, the pricing tier, LOCATION of the datacenter, and LEGAL TERMS, which basically give Microsoft permission to bill you for the MySQL database service.

STORAGE settings allows you to either create an additional storage account for the website to use or select an existing storage account.

After entering the configuration information, click CREATE. Azure will create the WordPress site for you. You can access it as an administrator and make any additional changes needed.

Configure and scale a website

Let's look at the configuration and scaling options for a website in the Azure Management Portal (*manage.windowsazure.com*). (Not all of the features are available in the Azure Preview Portal yet.) Log into the portal, select WEB SITES in the left column, and then click one of your websites.

Configuration

To find the configuration settings for the website, click the CONFIGURE option on the top of the page; see Figure 2-7.

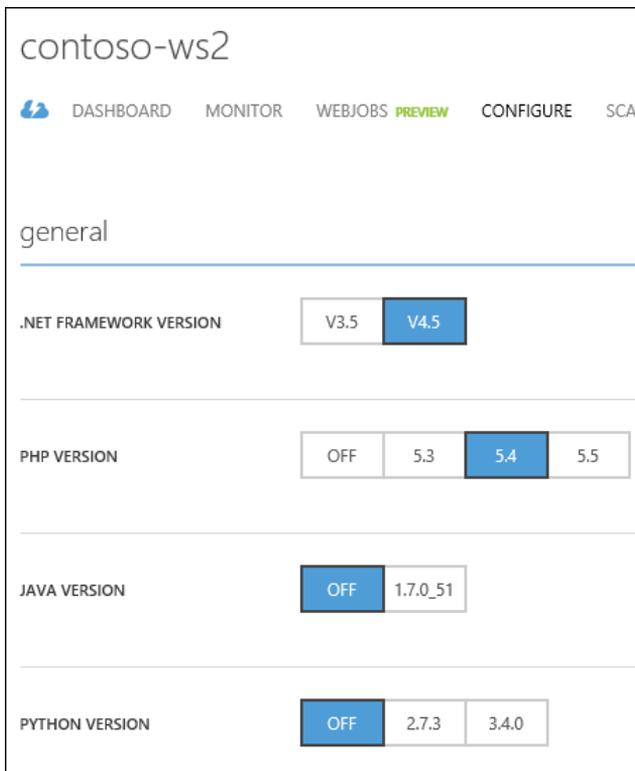


FIGURE 2-7 Configuration settings for the website.

This is the top of the CONFIGURE page. As you can see, you can set the versions for .NET, PHP, Java, and Python here. Scrolling farther down the page, we see additional general options for the website, as shown in Figure 2-8.

- **PLATFORM (32-BIT/64-BIT)** When you set up a Free website, this is set to 32-bit. Once you change your website to Standard, you might want to change this to 64-bit.
- **WEB SOCKETS (ON/OFF)** When this is enabled, you can use real-time request pattern applications that communicate using web sockets, such as chat.
- **ALWAYS ON (ON/OFF)** When this is enabled on a site, Azure will automatically ping your website regularly to ensure that the website is always active and in a warm/running state. Doing this will make sure that a site is always responsive and that the process or app domain has not paged out due to lack of external HTTP requests.
- **EDIT IN VISUAL STUDIO ONLINE (PREVIEW)** If you turn this on, a link to the editor will appear in the quick glance section of the DASHBOARD tab. This enables you to use Visual Studio Online to edit your website while it's live. If you do live editing and you have Deployment From Source Control enabled, if someone checks in a change it will overwrite the live changes you made.

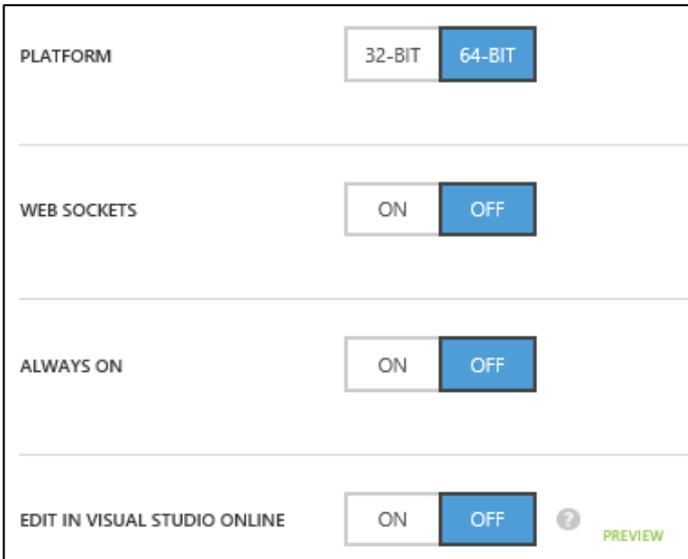


FIGURE 2-8 More general website options.

There are options for uploading certificates, managing domains, and managing your Secure Sockets Layer (SSL) bindings, as shown in Figure 2-9.

- **Certificates** You can upload an SSL certificate here. If you bind your SSL certificate to your custom domain name, end users can access your site using HTTPS.
- **Domain Names** This allows you to use a custom domain like mywebsite.contoso.com instead of mywebsiteatcontoso.azurewebsites.net.

- **SSL Bindings** This is where you bind the SSL certificate to the custom domain name.

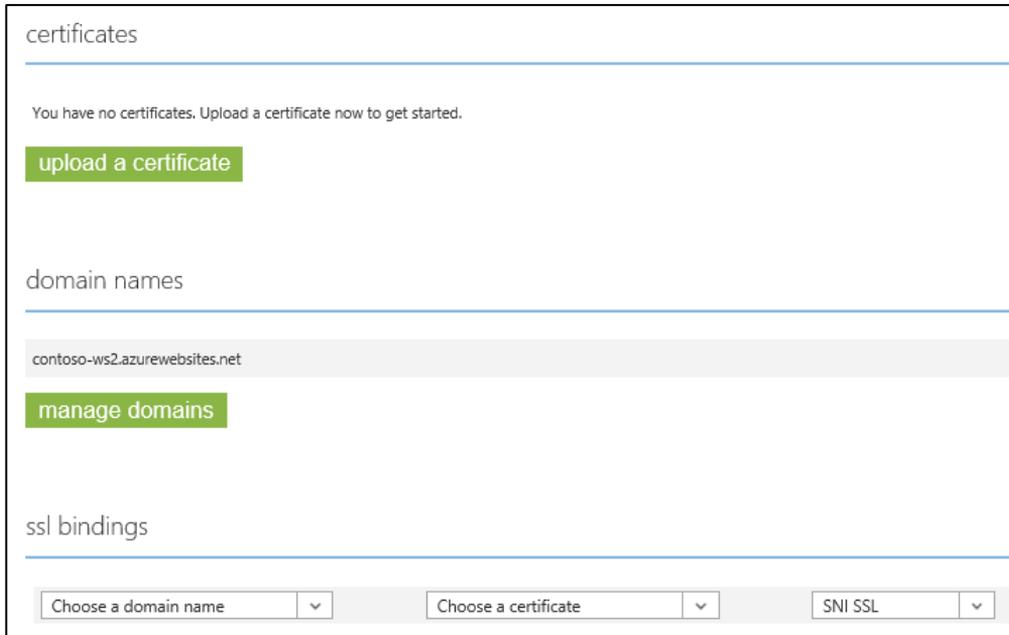


FIGURE 2-9 Manage certificates, domain names, and SSL bindings for the website.

The next section is used to configure application diagnostics; see Figure 2-10. To show as much as possible in the image, all of the settings are enabled.

- **APPLICATION LOGGING (FILE SYSTEM) (ON/OFF)** If this is turned on, then any logging performed by the web application will be written to the file system. You can access the logs by FTPing into the website. Because of the limited amount of disk space available, this will be enabled for 12 hours and then disable itself. The logging levels include Error, Warning, Information, and Verbose.
- **APPLICATION LOGGING (TABLE STORAGE) (ON/OFF)** If this is turned on, any logging performed by the web application will be written to Azure Tables. The logging levels include Error, Warning, Information, and Verbose. If you select this option, you will be prompted for the storage account and table to be used (see Figure 2-11). These logs are never deleted automatically.
- **APPLICATION LOGGING (BLOB STORAGE) (ON/OFF)** If this is turned on, it writes the logs to Azure Blob storage, storing logs for each hour in a separate blob. For these logs, you can specify a retention time in days; if you leave it blank, these logs will never be deleted automatically. If you select this option, you will be prompted for the storage account and container (Figure 2-12).

application diagnostics

Application tracing to the file system will be enabled for 12 hours.

APPLICATION LOGGING (FILE SYSTEM) ON OFF ?

LOGGING LEVEL

APPLICATION LOGGING (TABLE STORAGE) ON OFF ?

LOGGING LEVEL

[manage table storage](#)

APPLICATION LOGGING (BLOB STORAGE) ON OFF ?

LOGGING LEVEL

[manage blob storage](#)

SET RETENTION ?

RETENTION PERIOD days

FIGURE 2-10 Configuring application diagnostics for a website.

Manage Table Storage for Application Diagnostics

STORAGE ACCOUNT 

nightbirdstorage 

WINDOWS AZURE TABLE

Create a new table 

TABLE NAME

wawsapplogtablecontoso-ws2



FIGURE 2-11 Configuring Table storage for application diagnostics.

Manage Blob Storage for Application Diagnostics

STORAGE ACCOUNT 

nightbirdstorage 

WINDOWS AZURE BLOB CONTAINER

Create a new blob container 

BLOB CONTAINER NAME

wawsapplogblobcontoso-ws2 



FIGURE 2-12 Configuring Blob storage for application diagnostics.

The next section is used to configure site diagnostics (see Figure 2-13).

- **WEB SERVER LOGGING (OFF/STORAGE/FILE SYSTEM)** This indicates whether to write the web server (IIS) logs to Azure Tables or to the local file system. You can set the retention time if you choose STORAGE or FILE SYSTEM. For FILE SYSTEM, you also can set the QUOTA, or maximum amount of disk space the logs can take up, which must be between 25 MB and 100 MB.
- **DETAILED ERROR MESSAGES (ON/OFF)** This indicates whether to write summary error messages or detailed error messages.
- **FAILED REQUEST TRACING (ON/OFF)** This indicates whether to write the IIS Failure Logs.

site diagnostics

WEB SERVER LOGGING

QUOTA MB ?

SET RETENTION ?

RETENTION PERIOD days

DETAILED ERROR MESSAGES

FAILED REQUEST TRACING

FIGURE 2-13 Configuring site diagnostics.

In the next section, you can configure remote debugging (see Figure 2-14). If you turn this on and publish a debug version of your website, you can use Visual Studio to attach a debugger and debug your website while it's running in Azure.

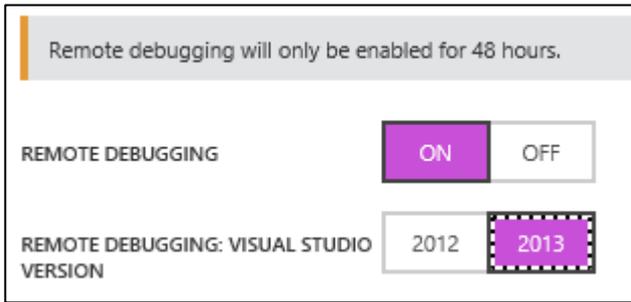


FIGURE 2-14 Configuring remote debugging.

In the next section, you can specify up to two endpoints to be monitored, as shown in Figure 2-15. Configuring this will allow you to monitor the availability of HTTP or HTTPS endpoints from up to three locations, including Chicago (IL), Amsterdam, Singapore, San Jose (CA), San Antonio (TX), Ashburn (VA), Hong Kong, and Dublin. If you have an internationally used application, this can help you pinpoint latency around the world.

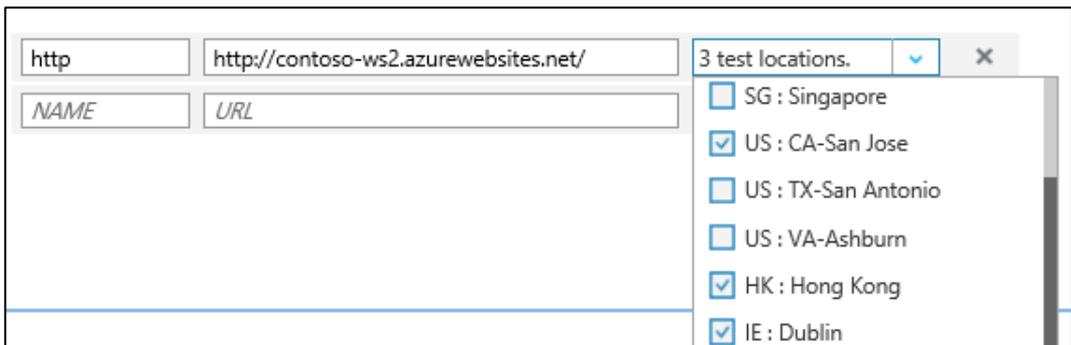


FIGURE 2-15 Configuring endpoint monitoring.

The only thing in the Azure Preview Portal that is not in the Azure Management Portal is the ability to set up and manage multiple deployment slots for your website, such as staging and production. This is under the Configuration section of the website blade in the Azure Preview Portal (portal.azure.com).

Scaling

You cannot scale a Free website; it is restricted to one instance. You can scale a Basic website manually up to three instances. For autoscaling, you must use a Standard website, which allows up to 10 instances. Let's look at the options using the Azure Management Portal (manage.windowsazure.com) because not all of the features have been migrated to Azure Preview Portal yet. First, we need to make sure the web hosting plan is STANDARD.

Log into the Azure Management Portal (manage.windowsazure.com), click WEBSITES in the left column, and then select the website you want to configure or autoscale. Click SCALE at the top of the

screen. You should see something similar to Figure 2-16.

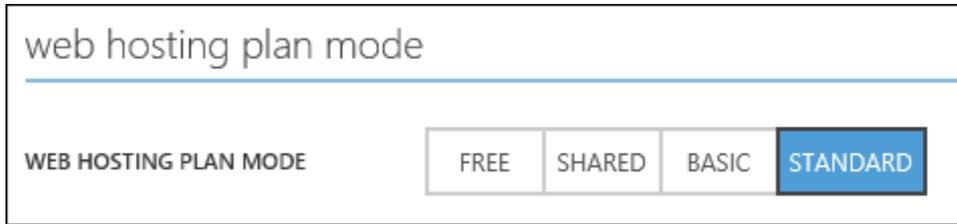


FIGURE 2-16 Web hosting plan.

Select STANDARD to change your plan and then click SAVE at the bottom of the screen. Note that you also can change the Instance Size (number of instances) on this screen.

The Azure Preview Portal has more details about the web hosting plan than the Azure Management Portal does. To check out those options, log into *portal.azure.com* and select your website, and then click WEB HOSTING PLAN in the actions on the top. If you do that now, be sure to come back to the Azure Management Portal to continue.

Now that we have a standard website, the scaling options are visible. First, you can scale according to a schedule. Clicking Set Up Schedule Times will display the entry screen shown in Figure 2-17.

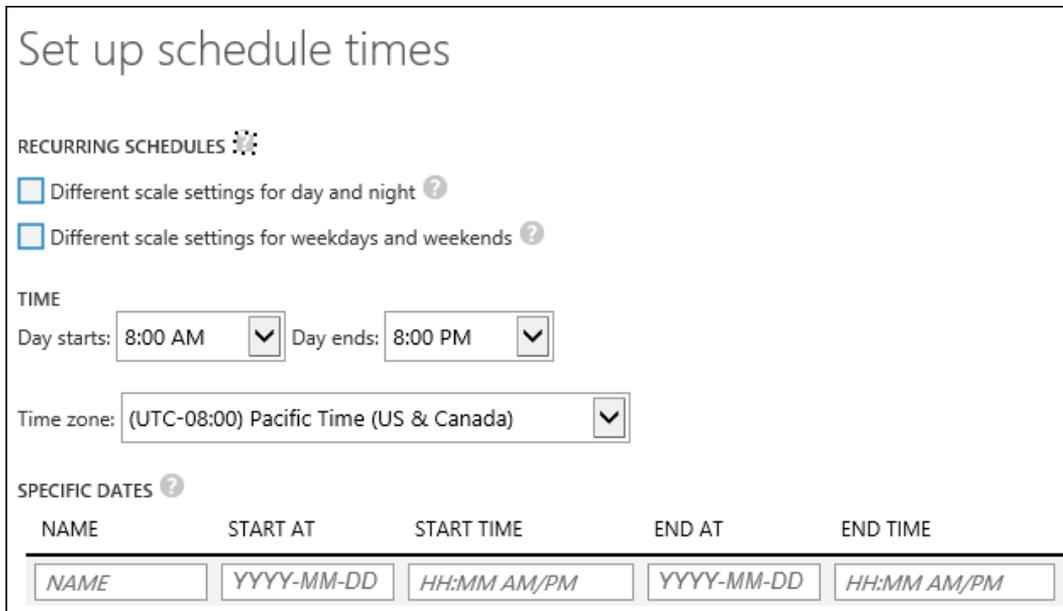


FIGURE 2-17 Scaling by schedule.

As you can see, there are several options. Additionally, you can set a default schedule and then override it for specific dates. For example, if you're scaling up from 8 A.M. to 5 P.M., you might want to

override that on a holiday.

You also can scale by CPU Percentage as shown in Figure 2-18.



FIGURE 2-18 Scale by metric.

This shows a chart of the number of instances for the past week. You can set the minimum and maximum number of instances (INSTANCE COUNT) and then set the TARGET CPU for the scaling. In this example, when the CPU hits 60 percent, it will increase the number of instances to three, and when it goes below 60 percent again, it will decrease them back to one. With Azure Websites, autoscaling takes about five minutes.

Deploying and monitoring websites

In this section, we look at options for creating website content, show how to publish your website from Visual Studio 2013, and look at the monitoring options in the Azure Management Portal.

Options for creating websites

There are multiple options for creating a website and pushing the content up to Azure Websites.

Notepad or an HTML editor

This is a pretty limited way to create a website, but if you're beginning on your journey to learn web development and want to create a simple HTML page, you can do that with Notepad or your favorite

HTML editing tool. After you're done, you can FTP to the website and transfer the files.

To FTP your files up to your website, log into the Azure Management Portal, click Websites, and then select your website. If you have not set up your login credentials yet, click Reset Your Deployment Credentials in the "quick glance" column. When prompted, provide a username and password. This is for access via Git or FTP.

Also under the "quick glance" column, the FTP HOST NAME and the DEPLOYMENT / FTP USER are provided. When you FTP in to your website, you will use these two pieces of information in addition to the password to access your website and place your files.

WebMatrix

This is a free, lightweight, cloud-connected web development tool that will enable you to create, publish, and maintain your websites. You can download this from <http://www.microsoft.com/web/webmatrix/>. Here are some of the features of this application:

- Seamless integration with Azure Websites.
- Can be used with PHP, Node.js, ASP.NET, HTML5, CSS3, and jQuery.
- Enables you to support many of the websites in the Website Gallery in the Azure Management Portal or the Website Marketplace in the Azure Preview Portal. Some examples of the websites available are Umbraco, WordPress, Joomla!, and Drupal.
- Allows management for SQL Server, SQL CE, and MySQL databases.
- Works seamlessly with Git and TFS.
- Can be used to develop websites locally or remotely using FTP or WebDeploy.

You can log in using the Microsoft account you use for Azure, create a new web application using one of the templates available, and publish it. When you log into either of the Azure Management Portals, you will see your new web application. You can make changes, verify the results in the local browser, and then republish. Republishing only deploys the modified files.

Visual Studio

Visual Studio is a full development environment, giving you the ability to create many different kinds of applications including, but not limited to, ASP.NET MVC applications, .NET client applications, Windows Communication Foundation (WCF) services, Web API, and Cloud Services, using languages such as C#, C++, VB, F#, and XAML.

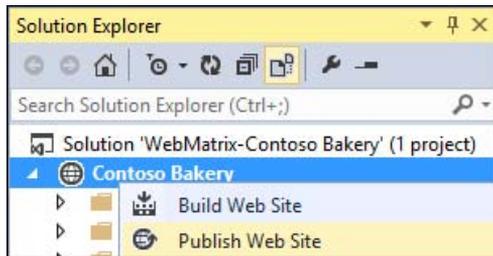
With Visual Studio, you can create a new web application and publish it directly to Azure Websites. We'll see how to do this in the next section.

Publishing a website from Visual Studio

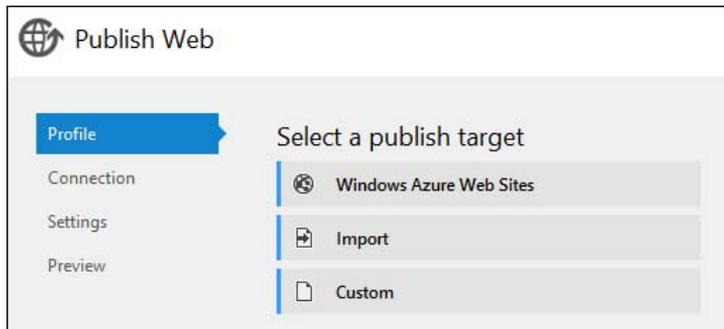
Open one of your web applications in Visual Studio. If you don't have a web application yet, create a new one with Visual Studio by selecting FILE > NEW PROJECT, selecting an ASP.NET Web Application, specifying the folder for the solution, and then selecting MVC Application. This gives you a basic MVC application that runs "as is." You can modify it later to make it your own.

Let's publish the web application to the Azure website we created earlier in this chapter.

1. Open your web application in Visual Studio. Right-click the website and select Publish Web Site.



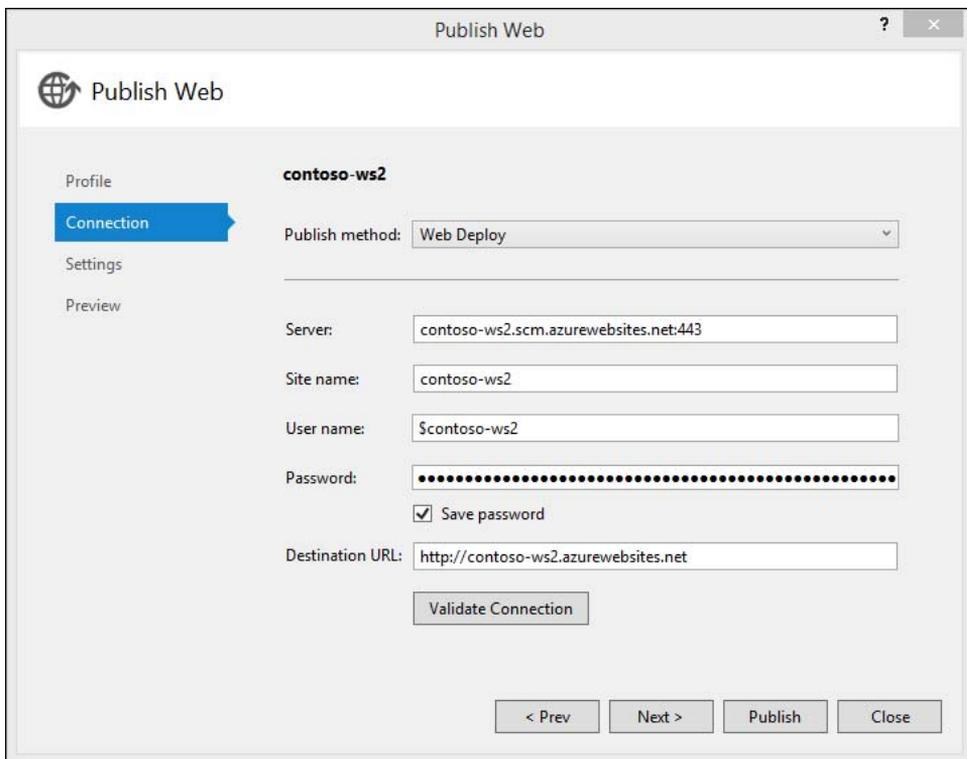
The Publish Web dialog will be displayed.



2. Select Windows Azure Web Sites. You will be asked to sign into your Azure account.
3. After signing in, you will be prompted to select the website to which to deploy. Select your website in the drop-down list and click OK. It retrieves the publishing settings from Azure and displays the connection information.



4. Click Validate Connection to make sure the connection works.



5. Click Next to go to the next screen, where you can select Debug or Release Configuration. Accept the defaults on that screen and click Next to go to the final screen.
6. On the final screen, you can preview the files that will be published. Click Publish to publish the website. All of the files will be deployed to the website, and then it will open the website.

When you make changes to your website and go through this same process to publish the website

again, it will only publish the files that have been added or modified.

Monitoring a website

You can set up many metrics to be monitored for a website. To do this, log into the Azure Management Portal (*manage.windowsazure.com*), select WEBSITES, and then select your website. When it brings up the Quick Start or the DASHBOARD, select MONITOR from the options at the top of the screen.

Initially, you will see only six metrics: CPU Time, Data In, Data Out, Http Server Errors, and Requests. If you set up Endpoint Monitoring on the CONFIGURE screen, you also will see the response times here (Figure 2-19).

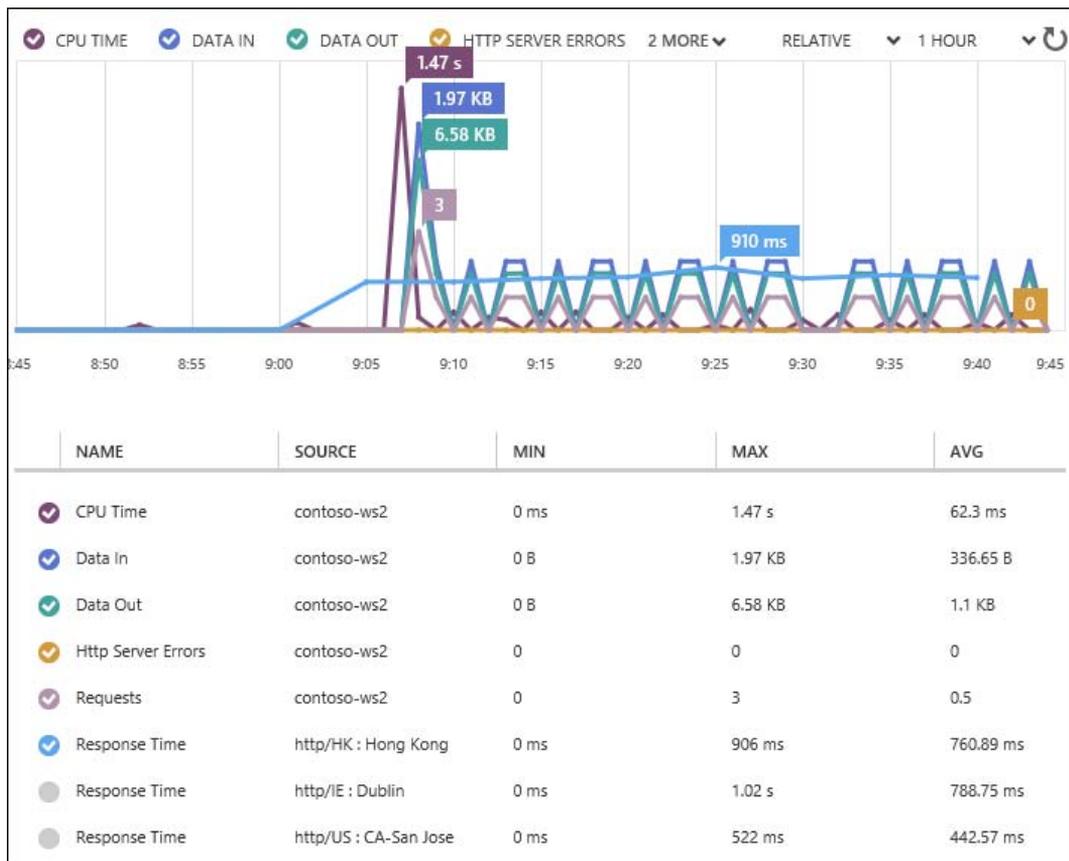


FIGURE 2-19 Monitoring a website.

You can see the endpoints defined earlier for Hong Kong, Dublin, and San Jose, CA. You can request as many metrics as you need to be displayed in the list, but you can select only six metrics to appear on the chart at any given time. You can add metrics by clicking +ADD METRICS at the bottom of the

screen. There are several more that can be selected.

The time frame displayed is 1 HOUR, as selected in the upper-right corner. Note that there is no y-axis. This is because each metric has its own y-axis value; it charts these to make the best of the space available.

There are other monitoring applications available through the Azure Store in the Azure Management Portal, including New Relic and App Dynamics. These can be selected and configured on the CONFIGURE screen in the Developer Analytics section.

PaaS Cloud Services

In Azure, there are two uses of the phrase “cloud service.” One is as a container for VMs that you create and maintain yourself. For example, you might create four VMs that are identical and put them in one cloud service. Then you would use the IP address of the cloud service as the entry point, and Azure would provide automatic load balancing of the four VMs. If you want to update the application running on those VMs, you must deploy it to each of the VMs. You can put them in an availability set to ensure that you always have a minimum number of VMs available. If you want to scale the application up or down, you must manually add or remove VMs to or from the cloud service or stop or start them. These VMs will be covered in Chapter 3.

The second kind of cloud service is one in which Azure maintains and manages your VMs for you. When you want to update all of the VMs, you just publish a new version of the application, and Azure updates each of the VMs, making sure to cycle through them so there’s no downtime. If you want to change the number of VMs, you just log into the Azure Management Portal and change the instance count, and Azure will add or remove the requested number of VMs for you.

This section is about the second kind of cloud service. In these cloud services, you have either web roles or worker roles. The only difference between the two is that web roles have IIS running in them by default. Web roles generally are used for web applications, WCF services, and anything else requiring IIS.

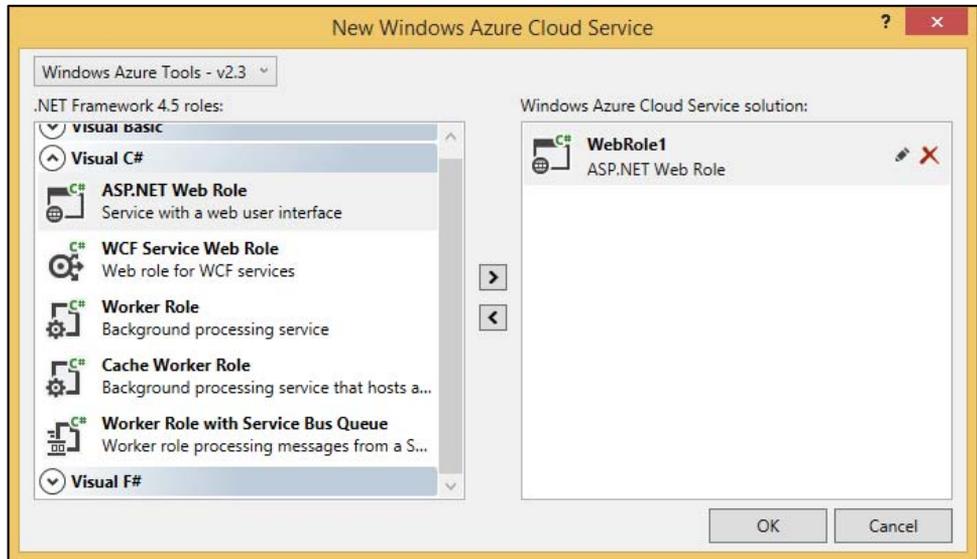
Worker roles generally are used for processing that needs to be continuous. For example, you might have a queue with messages on it that you want processed. The worker role runs an infinite loop that checks for messages on the queue and, if found, retrieves them and processes them. You can do substantial processing of images, video, files, etc. in a worker role.

Creating a cloud service

Let’s create a cloud service with a web role in it. Then we can look at the configurations and see how to publish it.

1. Using Visual Studio, create a new project (FILE > NEW PROJECT). For the type of project, select Windows Azure Cloud Service.

2. Fill in the name of the solution and click OK. (If you don't have Windows Azure Cloud Service in the list of projects, then you need to install the Azure SDK and Tools.) Next you will be prompted to select your role(s).
3. Select the ASP.NET Web Role and click the right-facing arrow to copy it to the right. Hover over it and click the pencil if you want to change the name of your web role before continuing and then click OK. Because you selected an ASP.NET Web project, you will be prompted to select which kind of ASP.NET project. Select MVC and click OK to continue.



Now when you look at your solution, you will see the project with the web role (MVC application) and the project with the cloud service; see Figure 2-20.

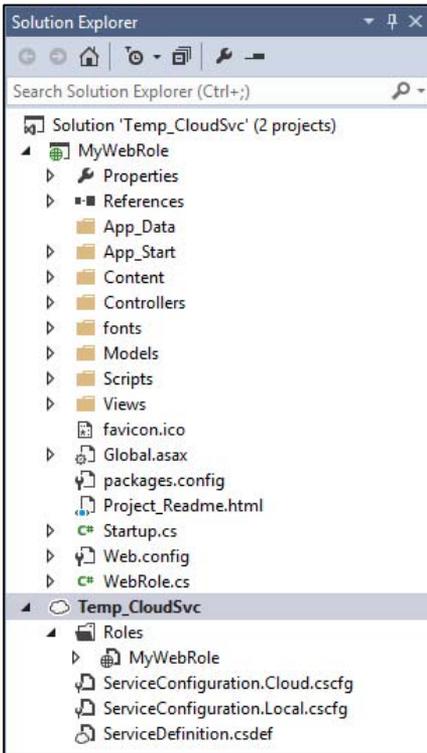


FIGURE 2-20 Web application and cloud service project.

In the web application project, notice there is a `WebRole.cs`. This is what starts up when the web role spins up in Azure. By default, this is blank, but you can put in event handlers for the role starting up and shutting down. For example, if you were going to use an Azure queue, you might want to add code to the role startup to make sure the queue exists and create it if it doesn't.

If you double-click the role, you can access its properties. Here are some of the most commonly used properties:

- **Configuration** VM Size, Instance count, diagnostics configuration, storage connection strings. To qualify for the service level agreement (SLA), a minimum instance count of two is required.
- **Settings** Configuration settings that you can retrieve in code. These can be modified in the Azure Management Portal while the site is running live. You can have settings for multiple configurations, including debug and release.
- **Endpoints** By default, a web role will have an HTTP endpoint open. You can modify that and open other endpoints such as HTTPS.
- **Local Storage** You can configure local storage for each instance. For example, if you wanted a 5 GB drive to use for temporary files, you would configure that here. Each instance would

have its own 5 GB local resource space.

- **Certificates** You can configure certificates to be used for SSL and for Remote Desktop (RDP) access here. You also must upload the certificate to the Azure Management Portal.

The settings are stored in XML format in the ServiceConfiguration.*.cscfg file(s). There are a couple of properties in the XML in that file that are not surfaced through the UI: osFamily and osVersion. osFamily selects the operating system that will be running in the VM. For example, osFamily 4 is Windows Server 2012 R2; osFamily 3 is Windows Server 2012; and osFamily 2 is Windows Server 2008 SP2.

To learn more about these values, check out the article "Azure Guest OS Releases and SDK Compatibility Matrix" at <http://msdn.microsoft.com/en-us/library/ee924680.aspx>. Unless your application requires a specific version, leave the values at the defaults, which are currently osFamily = "4" and osVersion = "*". The asterisk in osVersion means it will use the most recent version of the selected osFamily.

You can have multiple ServiceConfiguration.*.cscfg files. When you publish, it will ask which one you want to use. There is only one ServiceDefinition.csdef file. This has the master list of configuration setting variables and the endpoint definitions. It also has the instance size. Instance sizes range from Extra Small (shared, 1 CPU core, 768 MB memory) to A9 (16 CPU cores, 112 GB memory). Also available is the new D-series VM, with faster processors, a Solid State Drive (SSD) for the temporary disk, and a higher memory-to-core ratio.

For more details, check out "Virtual Machine and Cloud Service Sizes for Azure" at <http://msdn.microsoft.com/en-us/library/azure/dn197896.aspx>.

You can test a cloud service by using the storage and compute emulators. These are installed as part of the Azure SDK and Tools. If you make the cloud service your startup project, you can just hit F5 and the emulator(s) will start up and the application will run in your default browser.

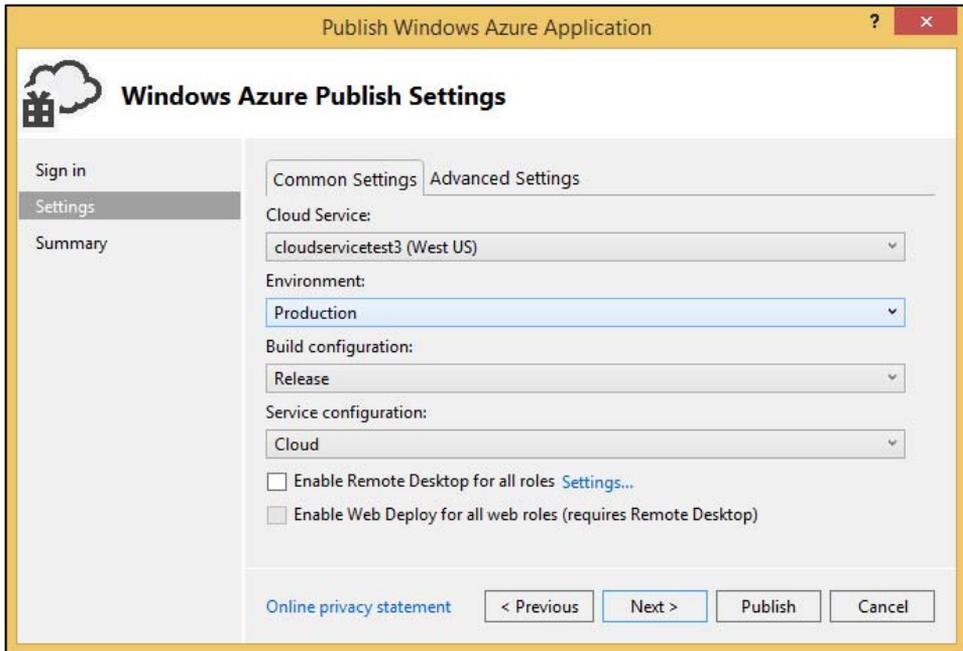
Publishing a cloud service

Let's publish our cloud service from the previous section.

1. Open the solution in Visual Studio.
2. Before publishing your cloud service for the first time, you have to create a cloud service in the portal. To do this, log into the Azure Management Portal (*manage.windowsazure.com*) and click CLOUD SERVICES in the left column.
3. On the bottom of the page, click +NEW > CLOUD SERVICE > QUICK CREATE; you will be prompted for URL and REGION. The URL must be unique among all cloud services. Note that the domain of the URL will be cloudapp.net. Select a region close to you and click CREATE CLOUD SERVICE. Azure will create your cloud service. This has no instances yet; when you publish your application, Azure will start up the instances, install your application, and then

enable them for access.

4. In Visual Studio, right-click the cloud service project and select Publish. You will be prompted for the Microsoft account you use for Azure; log in. Now it shows your subscription in the Publish Windows Azure Application window. Click Next to get to the Settings screen.



5. Select the cloud service you just added via the portal. For Environment, you can select Production or Staging (more on that later). If you are going to do remote debugging, select a Build Configuration of Debug; otherwise, set it to Release. The Service Configuration is a list of configurations from Visual Studio. This one shows Cloud and Local, but you could have one for production, one for staging, and one for development, and select which one to use here. This is a good way to support multiple environments with different configuration settings (such as database connection strings) in a cloud service.

The advanced settings are shown in Figure 2-21.

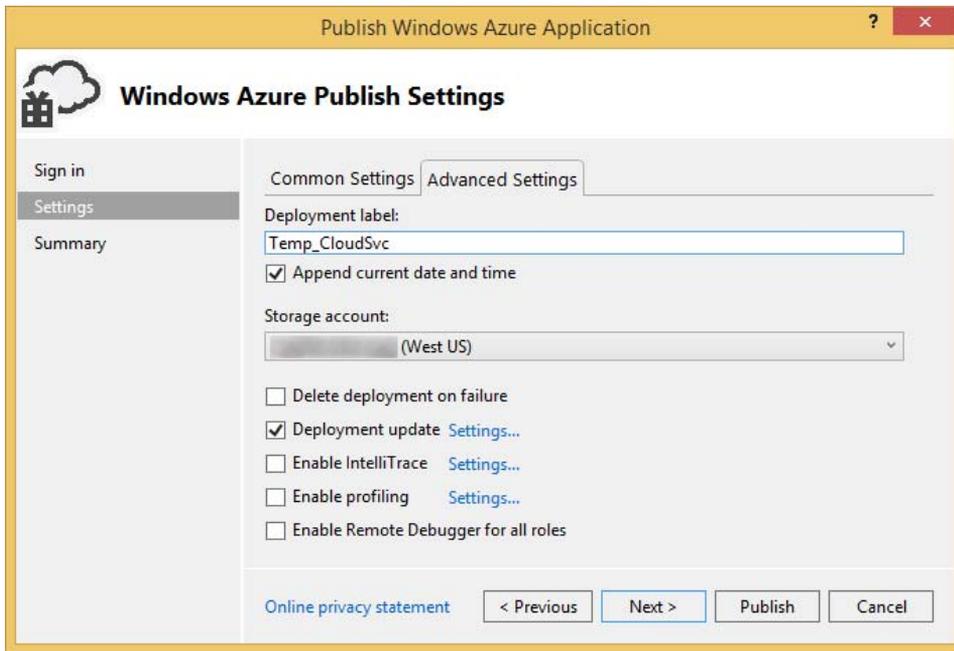


FIGURE 2-21 Advanced settings.

This is where you can set a deployment label. For example, you might put a version number in this field. You can also append the date and time to the deployment label. You will be able to see this in the portal, and it will help you know when it was last deployed.

The storage account is used to retain the package that is uploaded for you. The instances are built from the information in that package. The package consists of two files. One is the zipped versions of the application assemblies; the other is a configuration file specifying application configuration.

Accept the defaults here and click Next.

This will take you to a summary screen that will show all of your selections and let you save this publishing profile to be used again. After saving the publishing profile, click Publish.

Visual Studio will open a Windows Azure Activity Log window and display the progress of the deployment. It will verify the storage account, upload the package, create and start the instances with the right operating system on them, install your software, and then make the application available. This takes 5 to 10 minutes. The activity window will look similar to Figure 2-22 when publishing is completed.

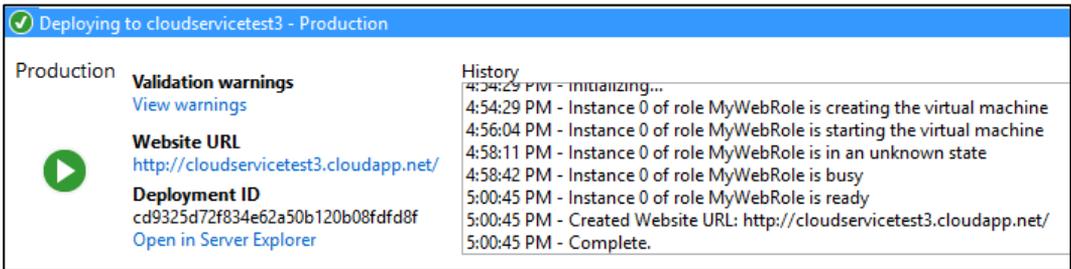


FIGURE 2-22 Publishing completed.

You can click the Website URL, and it will open your website in your default browser.

Another way to publish your cloud service is to right-click the cloud project and select Package and have Visual Studio create the deployment package for you. The package consists of two files. One is the zipped versions of the application assemblies; the other is a configuration file specifying application configuration.

Next you can go to the portal and do an UPDATE on PRODUCTION or select the STAGING environment and select UPLOAD. This will allow you to upload the package, and Azure will do the same thing as publishing from Visual Studio. It's just doing the upload package manually instead of having Visual Studio do it for you. You might want to do this if you have one group create the packages and another deploy them to staging or production.

Scaling and monitoring a cloud service

The easiest way to scale a cloud service is to set the VM Size and Instance count in the service configuration and republish. Another way to do this is to use the scaling options in the Azure Management Portal (*manage.windowsazure.com*). Log into the portal, and let's take a look at the features for the PaaS cloud service.

After logging in, click CLOUD SERVICES in the left column. Note that both kinds of cloud services are displayed here in the same list: the PaaS cloud services and cloud services that you create as a wrapper to one or more VMs, as discussed in the beginning of this section. Select the cloud service that you created and published in the previous section.

SCALE options

To scale a cloud service, go to the SCALE option. Here are the options available:

- SCALE BY SCHEDULED TIMES** To use this feature, click Set Up Schedule Times. This opens the schedule times screen. You can specify the start and end time for a day and scale settings for day versus night and weekdays versus weekends. You also can specify scale start and end times for specific dates. (This is identical to the Scale By Scheduled Times option for websites displayed in Figure 2-17.)

- SCALE BY METRIC: CPU or QUEUE** Several of the options are the same, whether scaling by CPU or QUEUE.

When scaling by CPU, you select the target CPU that will trigger the scaling event. For example, you might want it to scale when the CPU percentage is between 60 percent and 80 percent; see Figure 2-23.

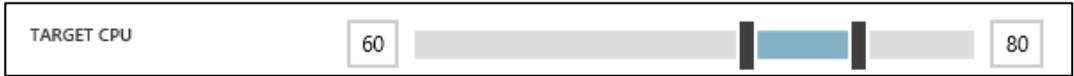


FIGURE 2-23 Setting target when scaling by CPU.

When scaling by queue, you select the storage account and queue name, as well as the target number of messages each instance can handle. If you take the total number of messages in the queue and divide it by the number of messages that each instance can handle, you get the number of instances that are needed. It will try to scale to this number or to the maximum number of instances, whichever is smaller. For example, if you set this to 500 and 2,000 messages come in, it will scale up to the maximum number of instances or four instances (2,000 messages divided by (500 messages/instance) = 4), whichever is smaller (see Figure 2-24).

FIGURE 2-24 Setting queue and queue target when scaling by queue.

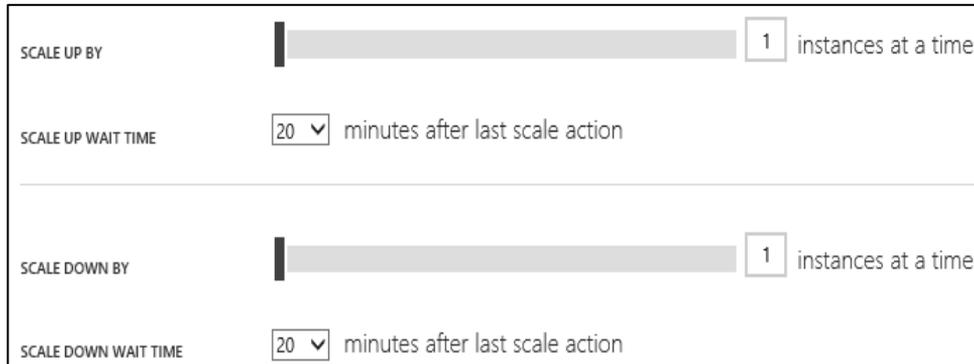
For both options, you can change the instance range. This is the minimum and maximum number of instances you want to have; see Figure 2-25.



FIGURE 2-25 Setting minimum and maximum number of instances.

For both options, you can select the number of instances by which to scale in or out with each autoscale event and how long to wait before scaling again. For example, you might want to scale out two instances at a time, but not more than once every 20 minutes. For scaling down, you might want

to scale in one instance at a time, but not more than once every 30 minutes in case the activity increases again; see Figure 2-26.



The image shows a configuration interface for Azure Autoscale. It is divided into two sections: 'SCALE UP' and 'SCALE DOWN'. Each section has a slider for the number of instances to scale and a dropdown menu for the wait time. The 'SCALE UP BY' slider is set to 1, and the 'SCALE UP WAIT TIME' dropdown is set to 20 minutes. The 'SCALE DOWN BY' slider is also set to 1, and the 'SCALE DOWN WAIT TIME' dropdown is set to 20 minutes.

SCALE UP BY	<input type="range" value="1"/>	1 instances at a time
SCALE UP WAIT TIME	<input type="text" value="20"/>	minutes after last scale action
SCALE DOWN BY	<input type="range" value="1"/>	1 instances at a time
SCALE DOWN WAIT TIME	<input type="text" value="20"/>	minutes after last scale action

FIGURE 2-26 Setting number of instances to scale up or down with each scaling action.

An important thing to note: the monitoring framework reads about 15 minutes behind and the data it is reading is a 45-minute average, so it takes about an hour for the autoscale event to be triggered. If the CPU spikes to 75 percent and comes right back down, it will not trigger an autoscaling event; it must be a sustained spike.

Monitoring

There are two ways to monitor performance for a cloud service. One is through the Azure Management Portal (MONITOR tab), and the other is through Visual Studio. In the portal, go to the MONITOR tab for the cloud service and click +METRICS on the bottom of the page; see Figure 2-27. There is a minimal list of metrics available.

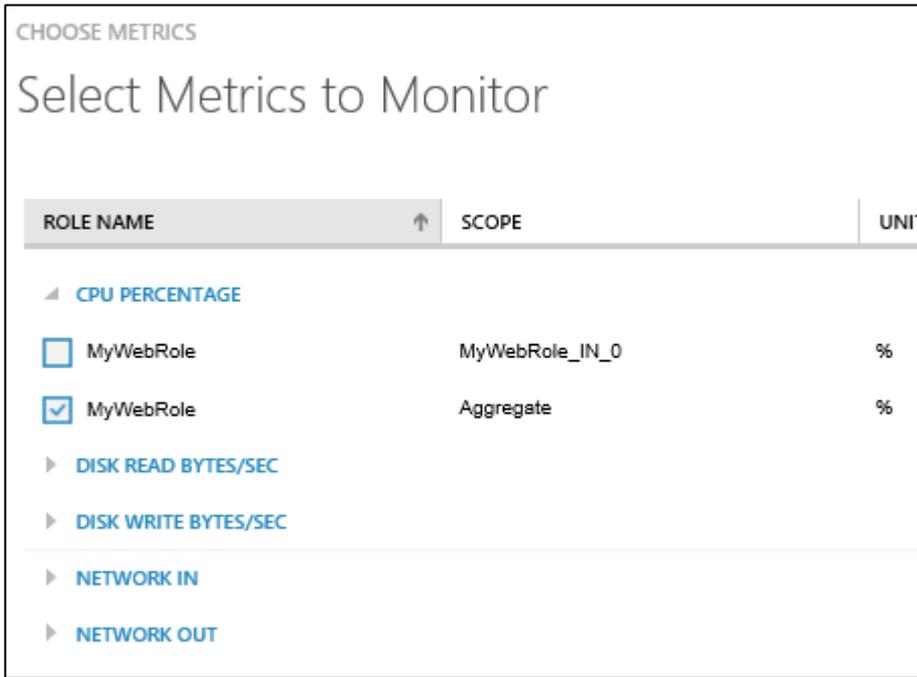


FIGURE 2-27 Selecting metrics to monitor.

If you go to the CONFIGURE tab and change the monitoring level to VERBOSE and let it run a bit, then click +METRICS again in MONITOR, you will find that many more performance metrics are available to add to MONITOR. Azure will write 5-minute, 1-hour, and 12-hour averages to tables in Azure Storage with names like WAD[deploymentid]PT1HRTTable. You can set the data retention time in days.

The other way to add performance metrics is through Visual Studio. Open your solution, go to the cloud service properties, and select the Configuration tab. You will find that you can define a Custom Diagnostics plan; see Figure 2-28.

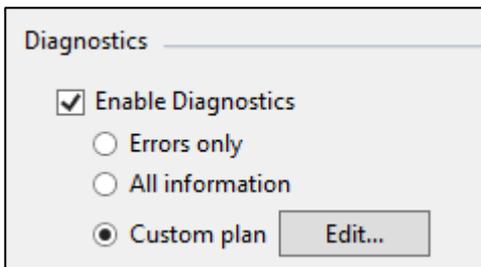


FIGURE 2-28 Enable diagnostics and choose to create a Custom plan.

Select Custom Plan and click Edit to open the Diagnostics configuration. On the Performance Counters tab, you will find many other metrics that you can add. You can even add performance counters not already in the list; see Figure 2-29.

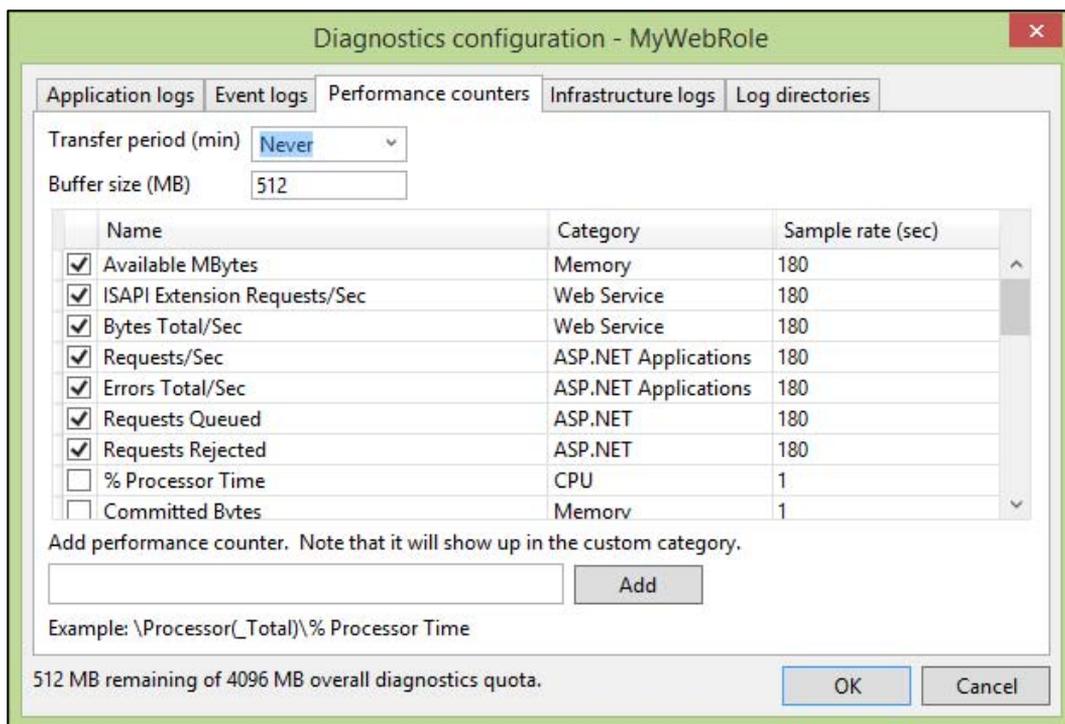


FIGURE 2-29 Adding performance counters using Visual Studio.

These diagnostics are written to the table called WADPerformanceCounterTable, and they are written roughly every five minutes as point-in-time values. There is no automatic retention time for this data; it will stay in the table until you remove it.

Of the two options, I recommend using the configuration in Visual Studio because the other tables have the deployment ID in the table name, and the deployment ID changes each time you deploy a cloud service. This means if you want to graph your metrics over time, you have to search each table. If you make the changes in Visual Studio, all of the metrics are stored in the same table, making it easier to query the information over time.

Miscellaneous points

Here a few items to keep in mind:

Configuration in the portal

You can modify the operating system family and version on the CONFIGURE tab in the Azure Management Portal. If you have configuration settings in your web role or worker role, you can edit those in the portal and save them. In both of these cases, the original package uploaded to Azure is not modified. This means if Azure installs patches on your VMs and reboots them, the manual changes you made through the portal will be lost. So if you make these changes via the portal, be sure to follow up by modifying the Visual Studio solution and deploying a new version.

Production and staging slots in the portal

Each cloud service has both a production and a staging slot. The production slot uses the URL assigned, such as *yourgreatwebapp.cloudapp.net*. The staging slot is assigned a globally unique identifier (GUID) for the URL, which changes each time you deploy a new package to it. This makes it difficult to test intertwined cloud services, such as A calls B calls C. You either have to change the address in A that points to B and the address in B that points to C, or you have to change the domain name system (DNS) entries, which can take time to propagate.

One way around this is to set up multiple cloud services for one web application in the Azure Management Portal. For example, you might set up MyWebApp for production and stMyWebApp for staging. Publish to staging and do your testing. When you are ready to put the changes in production, publish to the staging slot of MyWebApp and do a VIP Swap, which swaps the IP addresses for the two deployment slots, effectively putting the new one in production. Don't forget to delete the old one in the staging slot!

Worker roles

To add a worker role to your cloud project, right-click the Roles and select Add New Worker Role Project. Worker roles generally are used to retrieve messages from a queue and process them and then delete the messages from the queue.

A good example of this is having a web application that lets a customer upload pictures. You want to resize the pictures, so when the upload is completed, you write a message to the queue. The worker role would retrieve the message from the queue, resize the photos, put them in the destination folder, and then delete the message from the queue.

Chapter 3

Azure Virtual Machines

Platform-as-a-Service (PaaS) is without a doubt an attractive option for a certain category of workloads. However, not every solution can, or should, fit within the PaaS model. Some workloads require near total control over the infrastructure: operating system configuration, disk persistence, the ability to install and configure traditional server software, and so on. This is where Infrastructure-as-a-Service (IaaS) and Azure Virtual Machines come into the picture.

What is Azure Virtual Machines?

Azure Virtual Machines is one of the central features of Azure's IaaS capabilities, together with Virtual Networks. Azure Virtual Machines supports the deployment of Windows Server or Linux virtual machines (VMs) in a Microsoft Azure datacenter. You have total control over the configuration of the VM. You are responsible for all server software installation, configuration, and maintenance, as well as operating system patches.

Note The terminology used to describe the Azure Virtual Machines feature and a virtual machine instance can be a little confusing. Therefore, throughout this chapter, *Azure Virtual Machines* will refer to the feature, while *virtual machine* or *VM* will refer to an instance of an actual compute node.

There are two primary differences between Azure Cloud Services (that is, web and worker roles) and Azure Virtual Machines: control and persistence. As discussed in Chapter 2, "Azure Websites and Azure Cloud Services," PaaS cloud services consist of web and/or worker roles and are managed primarily by the Azure platform, allowing you to focus on creating the application and not managing the server infrastructure. With an Azure Virtual Machines VM, you are responsible for nearly all aspects of the VM.

Web and worker roles generally are considered stateless (largely due to the lack of a persistent disk), but Azure VMs are stateful servers and do feature persistent disks. There are two types of disks: an OS disk and a data disk. An OS disk is required, and the data disk is optional. More details on these disks will be provided later in this chapter, but for now understand the OS disk is where the operating system resides (Windows or Linux) and the data disk is where your application data resides.

Because of the level of control afforded to the user and the use of persistent disks, VMs are ideal for a wide range of server workloads that do not fit into a PaaS model. Server workloads such as database servers (SQL Server, Oracle, MongoDB, and so on), Windows Server Active Directory, SharePoint, and many more become entirely possible to run on the Microsoft Azure platform. If desired, users can move such workloads from an on-premises datacenter to one or more Azure regions, a process often called *lift and shift*.

Before continuing much further, it is important to understand a little more about what truly makes up an Azure cloud service. Fundamentally, a cloud service is a container for virtual machines. The container provides several key features, including a DNS endpoint, network connectivity (including from the public Internet if desired), security, and a unit of management. A cloud service can hold multiple VM types: Azure web and worker role instances or Azure Virtual Machines VMs. As of the time of this writing, a cloud service container cannot simultaneously contain both web and worker role instances and Azure Virtual Machines VMs (that is, a cloud service cannot host PaaS and IaaS VMs at the same time). As can be seen in Figure 3-1, an Azure cloud service contains either a collection of web and worker roles or VMs.

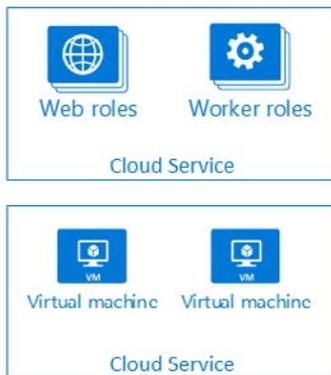


FIGURE 3-1 Cloud service container options.

VM status

Azure VMs have three possible states: Running, Stopped, and Stopped (Deallocated).

- **Running** The VM is on and running normally.
- **Stopped** The VM is stopped, but it is still consuming compute resources within Azure.
- **Stopped (Deallocated)** The VM is stopped, and it is not consuming compute resources within Azure.

By default, stopping a VM in the Azure Management Portal puts the VM into the Stopped (Deallocated) state. If you want to stop the VM but keep it allocated, you will need to use a PowerShell cmdlet:

```
> Stop-AzureVM -Name "az-essential" -ServiceName "az-essential" -StayProvisioned
```

Shutting down the VM from the operating system of the VM will also stop the VM but will not deallocate the VM.

IP address

The public virtual IP (VIP) address belongs to the cloud service, not the VM. Each VM has its own direct IP (DIP) address. As long as one or more VMs are Running or Stopped, the VIP will remain. If all the VMs are Stopped, the VIP will be released. If you want to stop (“power down”) the instance, yet preserve the VIP and DIP addresses, you will need to stop the VM, but do not deallocate the VM.

By default, VMs are assigned a dynamic IP address. If your use case requires the VM to have a static IP address, you can set the VM’s static IP address from within the Azure Management Portal or via PowerShell.

Billing

Azure Virtual Machines is priced on a per-hour basis, but is billed on a per-minute basis. For example, you are only charged for 23 minutes of usage if the VM is deployed for 23 minutes. The cost for a VM includes the charge for the Windows Server operating system. (Linux-based instances are slightly cheaper as there is no operating system license charge.) The cost, and the appropriate licensing, for any additional software you want to install is your responsibility.

There is a direct relationship between the VM’s status and billing:

- **Running** Billable
- **Stopped** Billable
- **Stopped (Deallocated)** Not billable

Create and configure virtual machines

There are two tiers for Azure Virtual Machines, Basic and Standard. VMs in the Basic tier are well suited for workloads that do not require load balancing or the ability to autoscale (more on those features later in this chapter). VMs in the Standard tier support all Azure Virtual Machines configurations and features.

Within the Basic and Standard tiers, there are various VM sizes available. The A-series VMs are the “traditional” sizes that have been around since Azure Virtual Machines was first introduced. The D-series VMs were introduced in September 2014, and they feature faster processors, a higher memory-to-core ratio, and a solid-state drive (SSD) for the temporary physical disk.

When you create a VM, you get two disks: an OS disk that is persisted in Azure blob storage and a temporary disk. The temporary disk is a physical disk located within the chassis of the server. The temporary disk (which is referred to as the D drive for Windows VMs) uses a traditional HDD platter for the A-series and SSD for the D-series VMs.

See Also For the most current Azure Virtual Machines configurations, please visit <http://msdn.microsoft.com/en-us/library/azure/dn197896.aspx>.

One of the easiest ways to get started creating Azure VMs is to use the Azure Management Portal or the Azure Preview Portal.

Note As of the time of this writing, there are two Azure management portals: the current Azure Management Portal at <http://manage.windowsazure.com> and the new preview version called the Azure Preview Portal at <http://portal.azure.com>. Unless otherwise indicated, examples in this chapter will be from the Azure Preview Portal.

Create a virtual machine with the Azure Preview Portal

If you haven't already done so, log into the Azure Preview Portal at <http://portal.azure.com>. At this point, you will need an Azure subscription. If you don't have one, you can sign up for a free trial at <http://azure.microsoft.com>.

To get started, click +NEW in the lower-left corner of the screen, and then the Everything label at the top of the New blade. As can be seen in Figure 3-2, doing so opens the Gallery blade, where you can then select the Virtual Machines option. You can now see the wide range of virtual machine options available from Microsoft and its partners. The images in the gallery include official images from Microsoft (for Windows systems), select partners such as Canonical and Oracle, and the OSS community via VMDepot. VMDepot provides a catalog of preconfigured Linux systems, applications, and development stacks you can use to deploy VMs in Azure. VMDepot images are not screened for security, compatibility, or performance. You can learn more about VMDepot at <http://vmdepot.msopentech.com>.

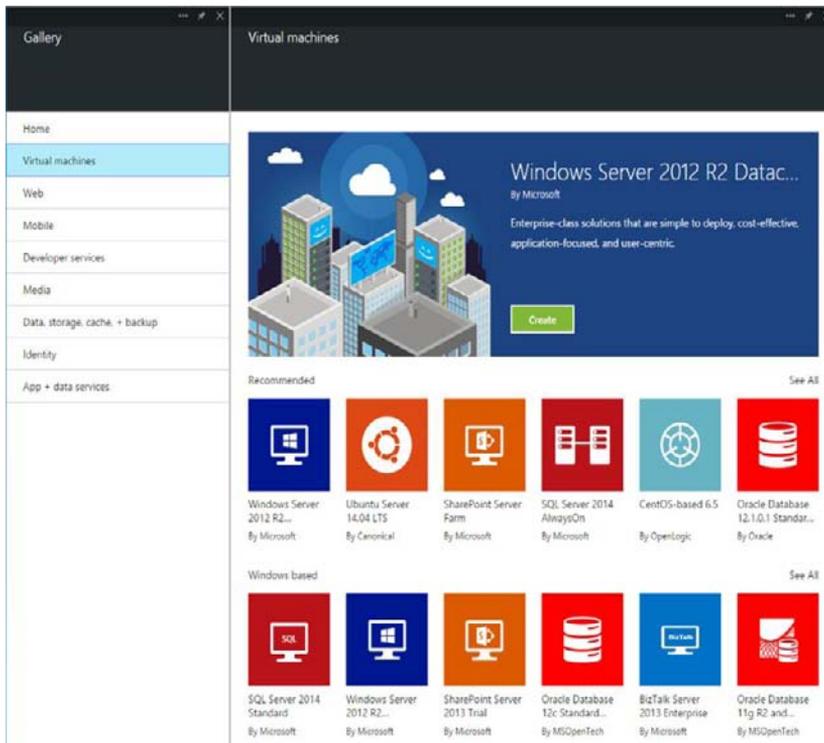


FIGURE 3-2 The Virtual Machines gallery.

For the purposes of this example, select the Windows Server 2012 R2 Datacenter image. On the resulting blade, you can read about the image, including any operating system updates. Click the blue Create button to proceed with creating your new VM.

Next, the Create VM blade should open. As you can see in Figure 3-3, on this blade you provide several important details about your new VM:

- **Host Name** The name of the VM.
- **User Name** The administrative user name.
- **Password** The password for the administrative user.
- **Pricing Tier** Expand this lens to view all the different pricing tiers (for example, Basic and Standard, A-series and D-series).
- **Optional Configuration** Expand this lens to control several important settings, such as:
 - The cloud service name (the DNS name; for example, contoso.cloudapp.net).
 - Whether operating system automatic updates (that is, Windows Update) is enabled (default is ON).

- The storage account to store the operating system disk's virtual hard drive (VHD).
- Any Virtual Network options (the VM will be placed in its own Virtual Network unless otherwise specified).
- Whether diagnostics should be enabled (default is not to enable diagnostics).
- **Resource Group** Provides a logical container for Azure resources (to help manage resources that are often used together).
- **Subscription** The Azure subscription to use if you have more than one.
- **Location** The Azure region where the VM should be placed.

The screenshot shows the 'Create VM' blade with the following configuration:

- HOST NAME:** contoso-vm1
- USER NAME:** contosoadmin
- PASSWORD:** masked with asterisks
- PRICING TIER:** Standard A0
- OPTIONAL CONFIGURATION:** Network, storage, diagnostics
- RESOURCE GROUP:** contoso-vm
- SUBSCRIPTION:** Windows Azure MVP MSDN Subsc...
- LOCATION:** East US

At the bottom, there is a checked checkbox for 'Add to Startboard' and a blue 'Create' button.

FIGURE 3-3 The Create VM blade.

When finished, click the blue Create button to instruct Azure to start creating your VM. This process could take several minutes.

Connect to a virtual machine

When creating a VM using the Azure Preview Portal, Remote Desktop is enabled by default. To connect to the VM, select the Connect button from the top of the desired VM blade, as shown in Figure 3-4.

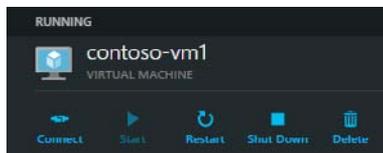


FIGURE 3-4 Connecting to a VM.

This will initiate a download to your local machine of a preconfigured Remote Desktop (.rdp) file. Open the RDP file and connect to the VM. You will need to provide the administrative user name and password set when initially provisioning the VM.

Configure disks

VHD files are used to create Azure Virtual Machines. There are two types of VHDs used in Azure Virtual Machines:

- **Image** A VHD that is a template for the creation of a new Azure VM. As a template, it does not have settings such as a machine name, administrative user, and so on. More information on creating and using images is provided later in this chapter.
- **Disk** A VHD that can be booted and used as a mountable disk for a VM. Once an image is provisioned, it becomes a disk. There are two types of disks: an OS disk and a data disk.

All persistent disks (the OS disk and data disks) are backed by page blobs in Azure Storage. Therefore, the disks inherit the benefits of blob storage: high availability, durability, and geo-redundancy options. Blob storage provides a mechanism by which data can be safely stored for use by the VM. The disks can be mounted as drives on the VM. The Azure platform will hold an infinite lease on the page blob to prevent accidental deletion of the page blob containing the VHD, the related container, or the storage account.

An OS disk is used precisely as the name suggests, for the operating system. For a Windows Server VM, the OS disk is the typical C drive; this is where Windows places its data. For a Linux VM, it is used for the /dev/sda1 partition used for the root directory. The maximum size for an OS disk is currently 127 GB for Windows.

The other type of disk used in Azure Virtual Machines is a data disk. The data disk is also used

precisely as the name would suggest, for storing a wide range of data. Each data disk has a maximum size of 1 TB. The data disks are often used for storing application data, such as data belonging to your custom application, or server software, such as SQL Server and the related data and log files.

Azure Virtual Machines also includes a physical temporary disk that is not persisted to Azure Storage. The temporary disk should be used only for temporary (or replicated) data, meaning the data will be lost in the event of a failure of the physical hardware. Figure 3-5 shows the various disk types.

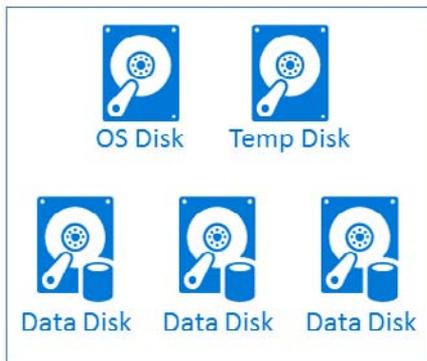


FIGURE 3-5 Disk types in Azure Virtual Machines.

Caching

Azure Virtual Machines has the ability to cache access to OS and data disks. Caching can potentially reduce transactions to Azure storage and can improve performance for certain workloads. There are three disk cache options: Read/Write, Read, and None.

The OS disk has two cache options: Read/Write (default) and Read.

The data disk has three cache options: Read/Write, Read, and None (default).

When creating a VM or attaching disks to an existing VM, be aware that there currently is a limit that no more than four data disks can have caching enabled.

Attach a disk

The VM created earlier in this chapter contains only a single disk, an OS disk. To add a data disk to the VM, you can start with a new, empty disk or upload a preexisting VHD. Either can be done using the Azure Preview Portal.

If you want to use an existing VHD, that VHD can be uploaded to Azure and attached to the VM. If you don't have an existing VHD, you can create one using Disk Management in Windows.

From the Disks lens, seen in Figure 3-6, you can see how many disks are attached to the current VM, along with the total size of all the attached disks.

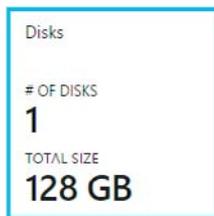


FIGURE 3-6 Number and size of disks.

To create and attach a new disk, first click on the Disks lens to open the Disks blade. On this blade, you will be able to attach a new disk or attach an existing disk.

To attach a new disk, click Attach New at the top of the Disks blade, as seen in Figure 3-7.

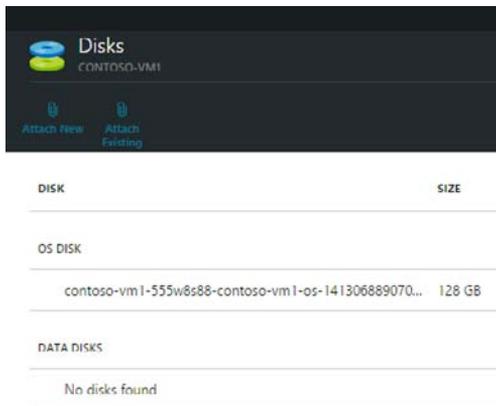


FIGURE 3-7 Attached disk details.

From the resulting Attach A New Disk blade, as seen in Figure 3-8, you will be able to provide several key settings:

- **Storage Container** The Azure Storage account and blob container that will store your new data disk.
- **Disk File Name** Provide your own or accept the default.
- **Size (GB)** The size of the new data disk (VHD).
- **Host Caching** The cache option to use for the data disk.

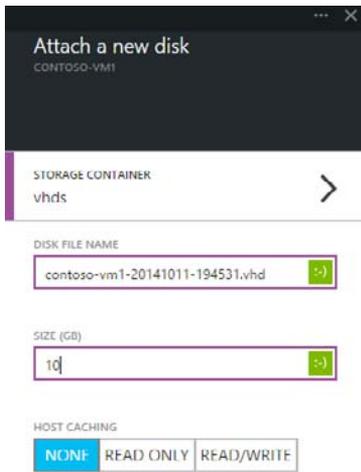


FIGURE 3-8 Attach a new data disk.

To attach an existing data disk, click **Attach Existing** at the top of the **Disks** blade. The resulting **Attach An Existing Disk** blade will present an option to select an existing VHD from your Azure Storage account, as you can see in Figure 3-9. You can use your favorite Azure Storage management tool to upload an existing VHD to a blob container in the desired storage account (be sure that VHD is set as a page blob and not a block blob).

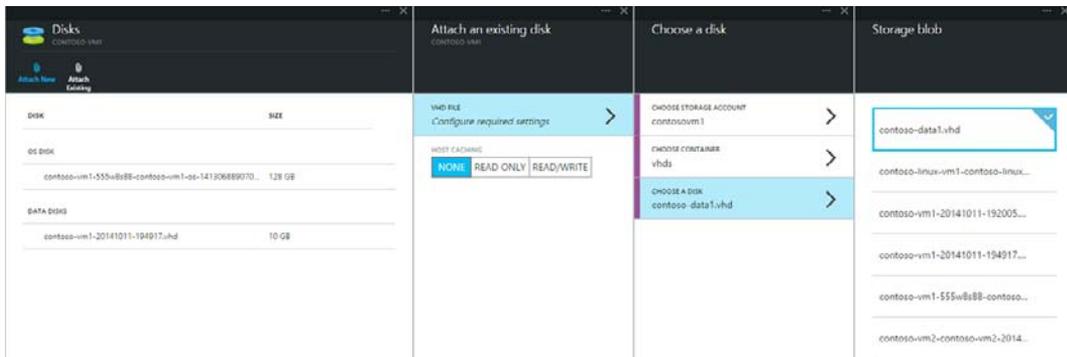


FIGURE 3-9 The Attach An Existing Disk blade.

Formatting disks

Once the data disks are attached to the Azure VM, each data disk needs to be formatted, just like a disk on a physical Windows server. Azure Storage uses page blobs to store the VHDs and does so using a sparse format. This means that Azure Storage charges apply only for data within the VHD that has actually been written. Because of this, it is recommended that you use a quick format when formatting the disks. A quick format will avoid storing large ranges of zeros with the page blob, thus conserving actual storage space and saving you money.

To format the disk(s), use Remote Desktop to remotely connect to the VM. Once connected and logged into the VM, open Disk Management. Disk Management is a native Windows application that allows you to view the disks and format any unallocated disks. As can be seen in Figure 3-10, proceed by right-clicking the unallocated disk and selecting Initialize Disk.

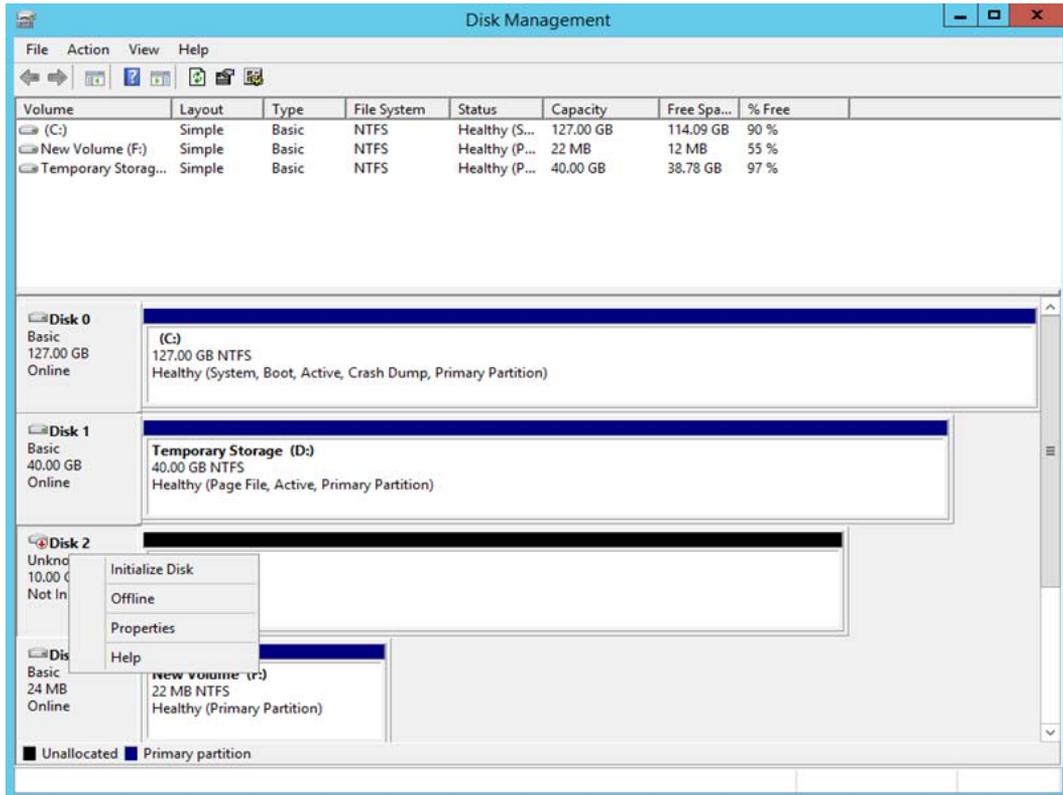
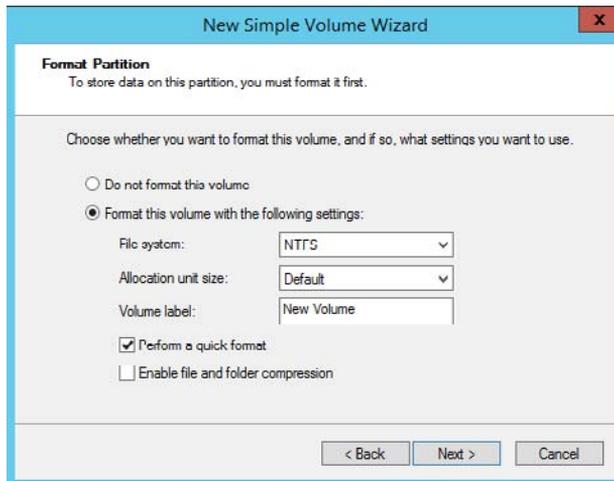


FIGURE 3-10 Windows Disk Management.

Complete the wizard to initialize the disk. Once the disk has been initialized, you can proceed with formatting the disk.

1. Right-click the disk and select New Simple Volume. The New Simple Volume Wizard should open.
2. Continue through the wizard, selecting the desired volume size and drive letter.
3. When presented with an option to format the volume, be sure to select Perform A Quick Format.



4. Finish the steps in the wizard to start formatting the disk.

Disk performance

Another factor to be aware of with Azure VM disks is IOPS. At the time of this writing, each data disk has a maximum of 500 IOPS and 60 MB/s (for Standard-tier VMs). This might or might not be sufficient for the desired workload. You should conduct performance tests to ensure the disk performance is sufficient. If it is not, consider striping the disk or using storage spaces (Windows Server 2012 or higher) as a way to increase disk performance.

See Also For more information on advanced configuration of Azure VM disks, including striping and storage spaces, please review the Microsoft Azure whitepaper available at <http://msdn.microsoft.com/en-us/library/azure/dn133149.aspx>. Although the referenced whitepaper is specific to running SQL Server on an Azure VM, the disk configuration details are common across a multitude of workloads.

Endpoints

The Azure Load Balancer exposes endpoints for an Azure cloud service. It is the configuration of the Azure Load Balancer that controls how requests from the Internet reach a specific port using a related protocol (such as TCP or UDP) on the VM. By default, Azure VMs are not able to accept requests from the Internet. To do so, a VM must be configured with one or more endpoints. This configuration is actually configuring the Azure Load Balancer to allow traffic from the Internet, creating a mapping between public ports on the Azure Load Balancer and private ports on the VM.

To view or edit, including adding endpoints for a VM, scroll down toward the bottom of the Virtual Machine blade and look for the Endpoints lens. There you will see the current endpoints, with their names and corresponding ports, as shown in Figure 3-11.

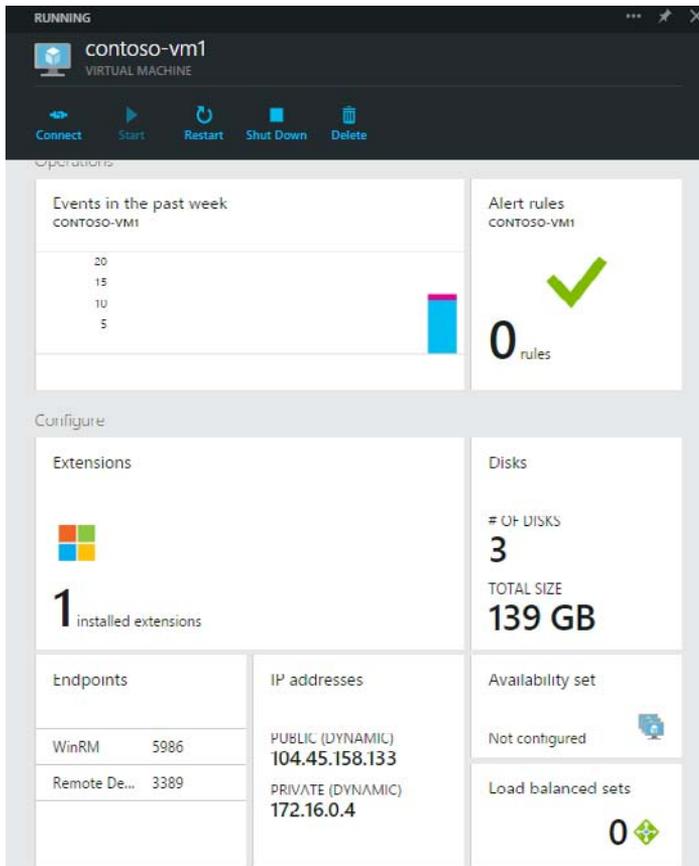


FIGURE 3-11 View current endpoints.

Additionally, it is possible to create a load-balanced configuration for a group of VMs. Doing so allows you to have multiple VMs work together, for example, as a collection of web servers in a web farm environment. With a load-balanced set, the Azure Load Balancer will distribute incoming requests across the available VMs instead of routing requests to a single VM.

As seen in Figure 3-12, selecting the Endpoints lens opens a new blade to view more details on the current endpoints, including the ability to remove the current endpoints or add new endpoints.

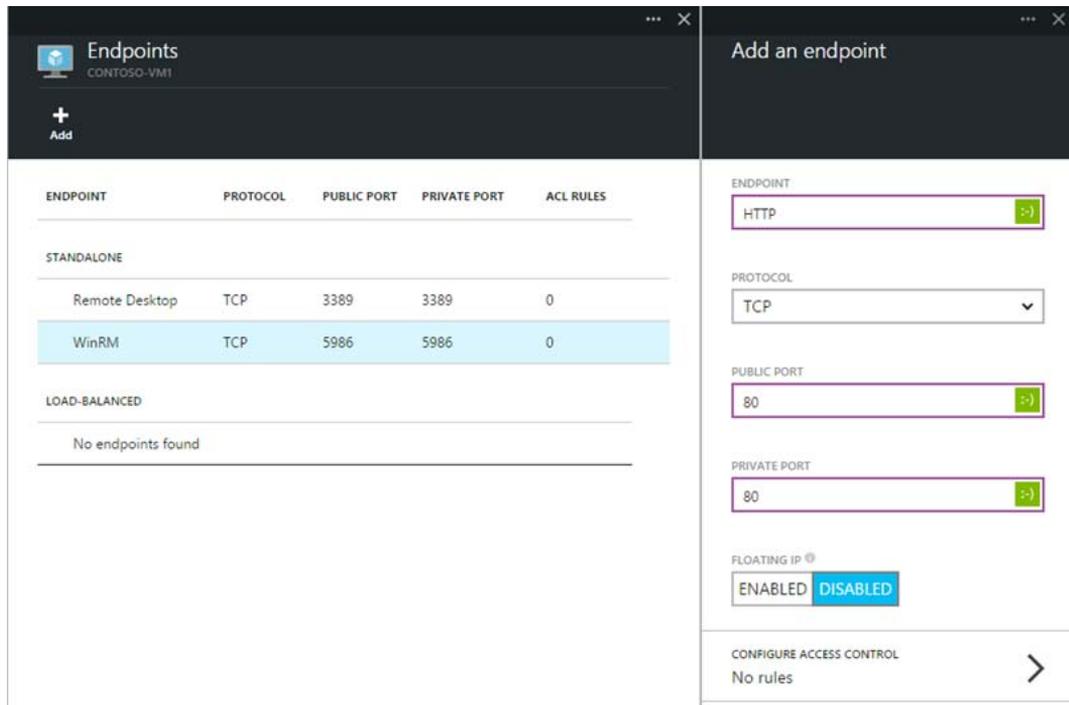


FIGURE 3-12 Add virtual machine endpoints.

Adding an endpoint instructs the Azure Load Balancer to allow traffic from the Internet using the specific protocol and port to reach your VM. If your VM is running its own firewall software, such as Windows Firewall for Windows Server instances, you might need to configure the server's firewall to allow traffic on the desired port and protocol.

Virtual machine management

Creating an Azure VM is only the beginning. There are several important factors that you should consider to successfully manage the VMs. Factors such as scalability, SLA, disk management, and machine maintenance are all important to consider.

The overall management of the VMs is largely the user's responsibility. The Microsoft Azure platform will ensure the VM is externally accessible. Other than that, you, the user, can do pretty much whatever you desire. Configuration and management of the VM can be done via a standard Remote Desktop connection or remotely using PowerShell.

Availability set

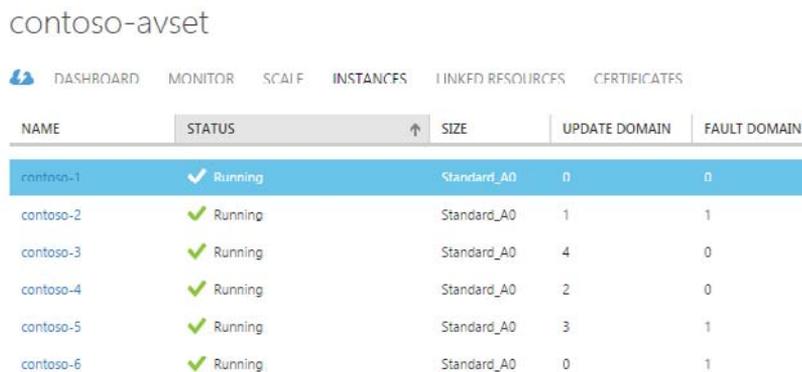
Azure VMs reside on physical servers hosted within Microsoft's Azure datacenters. As with most physical devices, there is a chance that there could be a failure. If the physical server fails, the Azure VMs hosted on that server will also fail. Should a failure occur, the Azure platform will attempt to quickly find a healthy host server on which to reconstitute the VM. This service-healing process could take several minutes. During that time, the application(s) hosted on that VM will not be available.

Besides hardware failures, the VMs could also be affected by periodic updates initiated by the Azure platform itself. Microsoft will periodically upgrade the host operating system on which the guest VMs are running (you're still responsible for the operating system patching of the guest VM that you create). During these updates, the VM will be rebooted and thus temporarily unavailable.

To avoid a single point of failure, it is recommended to deploy at least two instances of the VM. In fact, Azure provides an SLA only when two or more VMs are deployed into an availability set. This is a logical feature used to ensure that a group of related VMs are deployed so that they are not all subject to a single point of failure and so that they are not all upgraded at the same time during a host operating system upgrade in the datacenter. The first two VMs deployed in an availability set are allocated to two different fault domains, ensuring that a single point of failure will not affect them both simultaneously. Similarly, the first five VMs deployed in an availability set are allocated to five different update domains, minimizing the impact when the Azure platform induces host operating system updates one update domain at a time. VMs placed in an availability set should perform an identical set of functionality.

Sidebar: Virtual machine update and fault domains

You can view the update and fault domains used for your VMs by looking at the related cloud service in the Azure Management Portal at <http://manage.windowsazure.com>. As seen in Figure 3-13, the first five VMs are each placed in a different updated domain, and the sixth VM is placed in update domain 0.



contoso-avset

DASHBOARD MONITOR SCALE INSTANCES LINKED RESOURCES CERTIFICATES

NAME	STATUS	SIZE	UPDATE DOMAIN	FAULT DOMAIN
contoso-1	Running	Standard_A0	0	0
contoso-2	Running	Standard_A0	1	1
contoso-3	Running	Standard_A0	4	0
contoso-4	Running	Standard_A0	2	0
contoso-5	Running	Standard_A0	3	1
contoso-6	Running	Standard_A0	0	1

FIGURE 3-13 VMs, update domains, and fault domains.

If there is an existing availability set, the VM can be placed within the availability set as part of the VM provisioning process. If there is not an existing availability set, one will need to be created. To set the VM's availability set, select the Availability Set lens on the desired VM's blade, as seen in Figure 3-14.

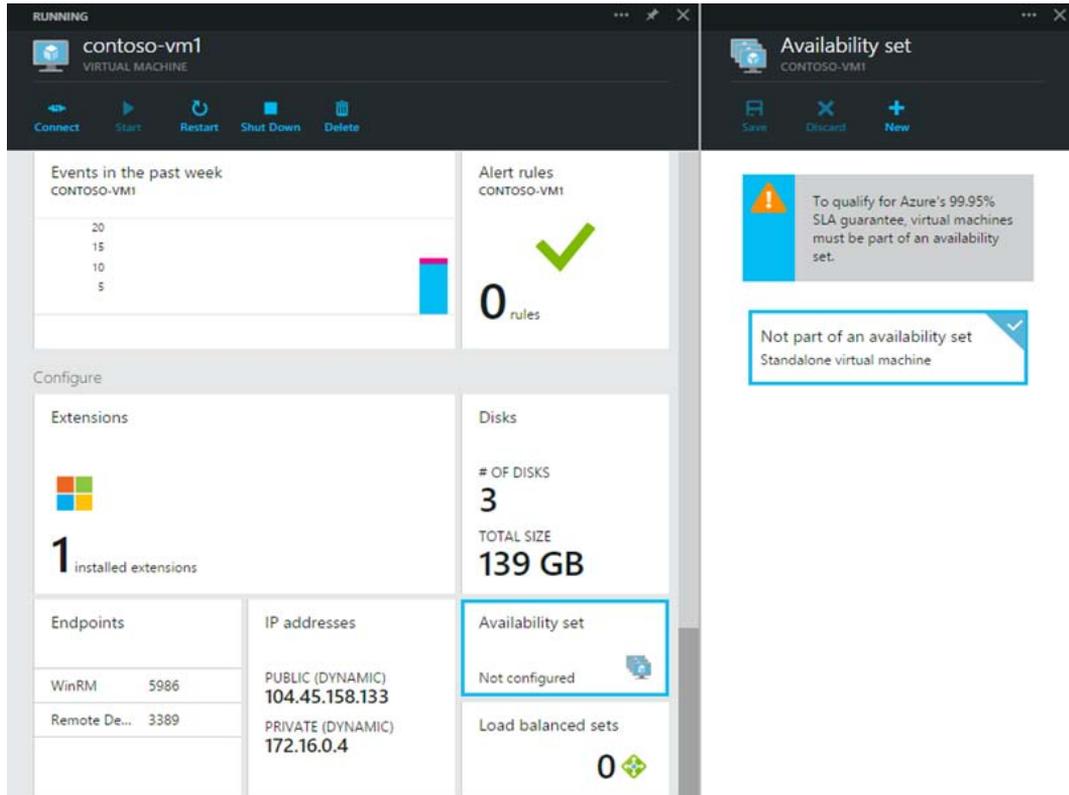


FIGURE 3-14 Beginning to create a new availability set.

Next, click **New** at the top of the Availability Set blade. On the resulting **New Availability Set** blade, provide the desired name for the new availability set.

If there is an existing VM not already in an availability set, it is possible to add the instance to an availability set. However, the VM must be part of the same cloud service (must have the same DNS name; for example, `contoso.cloudapp.net`). Select the desired VM and navigate to its Availability Set lens. From there, select the desired existing availability set. The VM will restart to apply the change.

Service level agreement

As of the time of this writing, Microsoft offers a 99.95 percent connectivity SLA for multiple instance VMs deployed in an availability set. That means that for the SLA to apply, there must be at least two instances of the VM deployed within an availability set.

See Also See the SLA at <http://azure.microsoft.com/en-us/support/legal/sla/> for full details.

Scalability

As with most Azure services, Azure Virtual Machines follow a scale out, not scale up, model. This means it is preferable to deploy more instances of the same configuration rather than adding larger, more powerful machines.

Before VMs can be scaled (out or in), the instances must be placed within an availability set. When determining the scale out approach for VMs, it is important to determine the maximum number of VMs, because that maximum number of VMs must be created, configured, and placed into an availability set. When it comes time to scale out, the VMs within the availability set are used to fulfill the scale-out needs. VMs within an availability set should all be the same size to take advantage of Azure's autoscale feature.

As of the time of this writing, to use Azure's autoscale features for VMs in an availability set, you will need to use the Azure Management Portal at <http://manage.windowsazure.com>. The ability to scale VMs will eventually be added to the Azure Preview Portal at <http://portal.azure.com>.

From within the Azure Management Portal, as seen in Figure 3-15, the Scale section provides the means to automatically scale out, or in, the number of VMs within a service. Scaling is an automatic operation handled by Azure based on provided configuration rules. The configuration can be related to time of day (for example, scale out during business hours and contract in the evening), or metrics such as average CPU utilization or queue depth (number of messages residing in the queue) for either Azure Storage queues or Service Bus. The average CPU utilization percentage is the average usage over the previous hour. For queue depth, the metric used is the target number of messages to be processed per machine, meaning Azure will determine the total number of messages in a queue divided by the number of instances when deciding if a scaling action should occur. By default, it will take approximately one hour for Azure to autoscale when the scale rules are met. This includes time during which VM metric data is collected and time necessary for a new VM to be provisioned. This means the default autoscale rules are often good for macrolevel scaling, but they likely are not suitable for rapid burst scale-out needs.

contoso-vm1-555w8s88

DASHBOARD MONITOR **SCALE** INSTANCES LINKED RESOURCES CERTIFICATES

summary

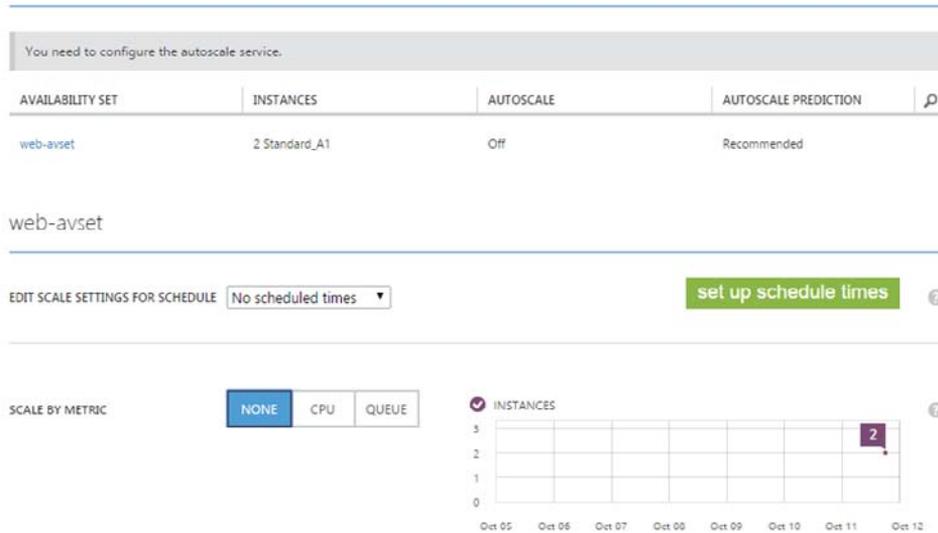


FIGURE 3-15 Autoscale settings.

See Also The 45-minute reaction time cannot be set via the Azure Management Portal, but it can be set via an API. (See the Azure Service Management API section for Operations on Autoscaling at <http://msdn.microsoft.com/en-us/library/azure/dn510374.aspx>).

The settings for scaling by CPU utilization will attempt to keep the average CPU utilization percentage across all instances within the defined range. As seen in Figure 3-16, you can control the target range of VMs and rules by which the VMs will scale out or in. You can set the minimum and maximum number of VMs that can be used, thus ensuring that you will never have too few or too many VMs. Should a scaling action be necessary, Azure will scale in or out based on the number of VMs defined for when a scaling action should occur. Be aware that there is a wait time, or “cooldown” period, that is applied after each scaling action. This prevents the system from thrashing VMs up and down and provides the system a period to attempt to stabilize before taking another scaling action.

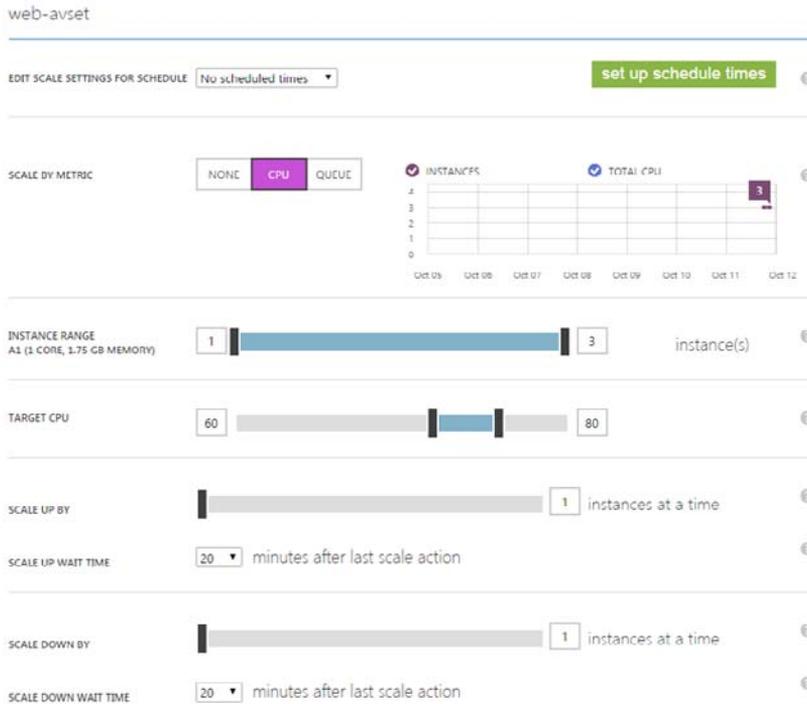


FIGURE 3-16 Scale by CPU.

As seen in Figure 3-17, the settings for queue utilization are largely the same as those for CPU utilization. Instead of making a scale decision based on average CPU utilization, the metric to use is the number of messages in a specific Azure Storage or Service Bus queue. More accurately stated, it is the target number of messages that a VM should process. Azure will automatically adjust the number of VMs based on the total number of messages divided by the target messages per VM. For example, if there are 2,000 messages in the queue, and the target is 500 messages per VM, Azure will attempt to scale up to four VMs, following the scale-up rule and wait time.

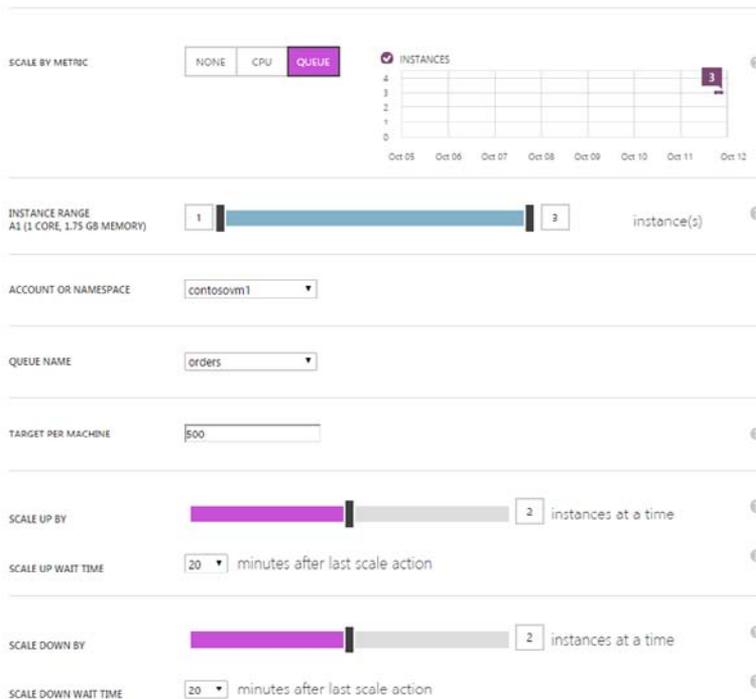


FIGURE 3-17 Scale by queue.

Keep in mind that the service can only scale out until the core limit is reached on the subscription. This means if the subscription has a limit of 20 cores, the VMs being deployed cannot in total consume more than 20 cores.

Image capture

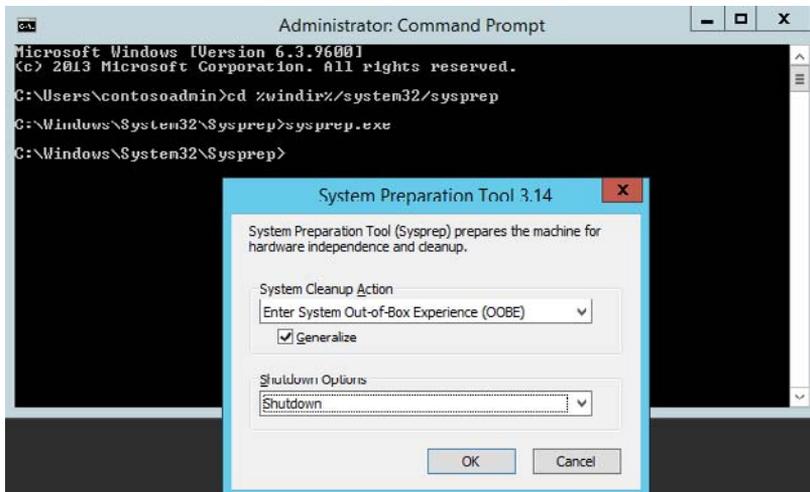
Once you have your new Azure VM configured as you would like it, you might find yourself wanting to create a clone of the VM. For example, you might want to create several more VMs using the one you just created as a template. This process is referred to as capturing the VM, or creating a generalized VM Image. When you create a VM Image, you capture not only the OS disk, but also any attached data disks.

When you capture the VM to use it as a template for future VMs, you will no longer be able to use the original VM (the original source) because it is deleted after the capture is completed. Instead, you will find a template image available for use in your Virtual Machine gallery in the Azure Management Portal.

There are several steps you will need to take to capture a VM so it is available for use as a template image. As of the time of this writing, capturing VMs is only supported in the Azure Management Portal at <http://manage.windowsazure.com>. The following steps refer to capturing a VM running Windows

Server. If you're running a Linux VM, the process would be similar except you would execute **waagent -deprovision** instead of a Windows sysprep command.

1. Connect to the VM using Remote Desktop (as discussed earlier in this chapter).
2. Open a command prompt window as the administrator.
3. Navigate to the `%windir%/system32/sysprep` directory and then run `Sysprep.exe`.
4. In the System Preparation Tool, perform the following actions:
 - a. From the System Cleanup Action list, select Enter System Out-Of-Box Experience (OOBE).
 - b. Select the Generalize check box.
 - c. In Shutdown Options drop-down list, select Shutdown.



5. The VM will run sysprep. If you are still connected to the VM via RDP, you will be disconnected when it begins to shut down. Watch the VM in the Azure Management Portal until it completely shuts down and shows a status of "Stopped."
6. From the Azure Management Portal, select the VM to capture and then click the CAPTURE button in the bottom tray menu.

INSTANCES IMAGES DISKS

NAME	STATUS	SUBSCRIPTION	LOCATION
contoso linux vm1	Stopped (Deallocated)	Windows Azure MVP MSDN Subs...	East US
contoso-vm1	Stopped	Windows Azure MVP MSDN Subs...	East US
contoso-vm5	Stopped (Deallocated)	Windows Azure MVP MSDN Subs...	East US



7. Provide a name for the new image and select the I Have Run Sysprep On The Virtual Machine check box.

×

Capture the virtual machine

IMAGE NAME
contoso vm1 2036 023612

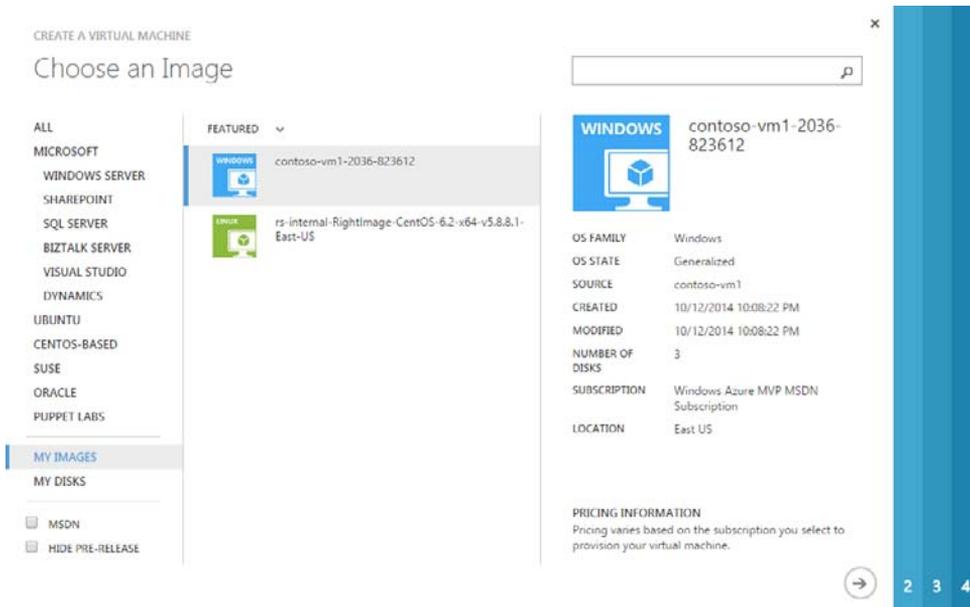
IMAGE DESCRIPTION (LABEL)
My Great VM

I have run Sysprep on the virtual machine ?

The virtual machine will be deleted after the image is captured.

✓

8. Click the check mark to capture the image.
9. The image should now appear in your Virtual Machine gallery, under My Images. You can now use this image to create a new VM instance.



There is an alternative to the process just described. If you would like to create a snapshot of the VM for any reason (for example, you are about to install software you are not confident will work), you can create a specialized VM Image. A specialized VM Image, like a generalized VM Image, contains the OS disk and all attached data disks. The disks are treated as read-only disks and copied when deploying the new VM. You can create a specialized VM Image just by selecting the desired VM in the Azure Management Portal and clicking CAPTURE. There is no need to go through the sysprep process. Although you can capture a specialized VM Image while the machine is running, you do so at the risk of losing any data in memory.

Chapter 4

Azure Storage

Microsoft Azure Storage is a Microsoft-managed service that provides durable, scalable, and redundant storage. Microsoft takes care of all backups and maintenance for you. An Azure subscription can host 50 storage accounts, each of which can hold 500 TB. Azure Storage provides various types of services: Blob service, File Share service, Table service, and Queue service. You can turn on metrics for each storage account, and you can set alerts to notify you when one of the metrics becomes excessive (such as data egress).

In this chapter, we take a look at those four services in Azure Storage. We talk about each one, discuss what they are used for, and, in some cases, show how to manage them.

Just announced at the time of this writing is a new feature called Premium Storage. This is a new type of SSD-based storage, designed to support I/O intensive workloads. More information is available at <http://azure.microsoft.com/blog/2014/10/20/azures-getting-bigger-faster-and-more-open/>.

Blob storage

The word *blob* is an acronym for binary large object. Blobs are basically files like those that you store on your computer (or tablet, mobile device, etc.). They can be pictures, Microsoft Excel files, HTML files, virtual hard drives (VHDs)—pretty much anything.

The Azure Blob service gives you the ability to store files and access them from anywhere in the world by using URLs, the REST interface, or one of the Azure SDK storage client libraries. (Storage client libraries are available for multiple languages, including .NET, Node.js, Java, PHP, Ruby, and Python.) To use the Blob service, you have to create a storage account. Once you have a storage account, you can create containers, which are similar to folders, and then put blobs in the containers. You can have an unlimited number of containers in a storage account and an unlimited number of blobs in each container, up to the maximum size of a storage account, which is 500 TB. The Blob service supports only a single-level hierarchy of containers; in other words, containers cannot contain other containers.

Azure Storage supports two kinds of blobs: block blobs and page blobs. Block blobs are used to hold ordinary files up to 200 GB in size. The primary use case for block blobs is the storage of files that are read from beginning to end, such as media files or image files for websites. They are named *block blobs* because files larger than 64 MB must be uploaded as small blocks, which are then consolidated (or committed) into the final blob. Page blobs are used to hold random-access files up to 1 TB in size. The primary use case for page blobs is as the backing storage for the VHDs used to provide durable disks for Azure Virtual Machines, the IaaS feature in Azure Compute. They are named *page blobs* because they provide random read/write access to 512-byte pages.

Blobs are addressable through a URL, which has the following format:

http://[storage account name]/blob.core.windows.net/[container]/[blob name]

The Blob service supports only a single physical level of containers. However, it supports the simulation of a file system with folders within the containers by allowing blob names to contain the '/' character. The client APIs provide support to traverse this simulated file system. For example, if you have a container called animals and you want to group the animals within the container, you could add blobs called cats/tuxedo.png, cats/marmalade.png, and so on. The URL would include the entire blob name including the "subfolder," ending up with something like this:

http://mystorage.blob.core.windows.net/animals/cats/tuxedo.png

http://mystorage.blob.core.windows.net/animals/cats/marmalade.png

When looking at the list of blobs using a storage explorer tool, you can see either a hierarchical directory tree or a flat listing. The directory tree would show cats as a subfolder under animals and would show the .png files in the subfolder. The flat listing would list the blobs with the original names, cats/tuxedo.png and cats/marmalade.png.

You also can assign a custom domain to the storage account, which changes the root of the URL, so you could have something like this:

http://[storage.companyname.com]/[container]/[blobname]

This eliminates cross-domain issues when accessing files in blob storage from a website, because you could use the company domain for both. Blob storage also supports Cross-origin resource sharing (CORS) to help with this type of cross-source usage.

Azure Files (preview)

The Azure Files service enables you to set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. This means that multiple virtual machines (VMs) can share the same files with both read and write access. The files also can be accessed by using the REST interface or the storage client libraries. The Files service removes the need for you to host your own file share in an Azure VM and go through the tricky configuration required to make it highly available.

This can be used for many common scenarios:

- Many on-premises applications use file shares; this makes it easier to migrate those applications that share data to Azure.
- Configuration files can be stored on a file share and accessed by multiple VMs.
- Diagnostic logs, metrics, crash dumps, etc. can be saved to a file share to be processed and

analyzed later.

- Tools and utilities used by multiple developers in a group can be stored on a file share to ensure that everyone uses the same version and that they are available to everyone in the group.

To make the share visible to a VM, you just mount it as you would any other file share, and then you can access it through the network URL or the drive letter to which it was assigned. The network URL has the format `\\[storage account name].file.core.windows.net\[share name]`. After the share is mounted, you can access it using the standard file system APIs to add, change, delete, and read the directories and files.

At the time of this writing, the Azure Files Preview is not surfaced in the Microsoft Azure Preview Portal at all, and it is only surfaced in the Microsoft Azure Management Portal to confirm that the feature is enabled. To do this, log into the Azure Management Portal and go to the storage account—the endpoint URL will be displayed with the `files.core.windows.net` domain.

To create or view a file share or upload or download files to it from outside Azure, you must use PowerShell, the REST APIs, one of the storage client libraries, or AzCopy, a command-line tool provided by Microsoft. (For more information on AzCopy, check out this link: <http://azure.microsoft.com/en-us/documentation/articles/storage-use-azcopy/>.)

This is a preview feature, so there are some limitations:

- When using SMB 2.1, the share is available only to VMs within the same region as the storage account. You can, however, access the data from anywhere by using the REST APIs.
- The storage emulator does not support Azure Files.
- The file shares can be up to 5 TB.
- Throughput is up to 60 MB/s per share.
- The size limit of the files placed on the share is 1 TB.
- Up to 1,000 IOPS (of size 8 KB) per share.
- Active Directory–based authentication and access control lists (ACLs) are not currently supported, but it is expected that they will be supported at some time in the future. For now, the Azure Storage account credentials are used to provide authentication for access to the file share.

For files that are accessed repeatedly, it might be preferable to split a set of files among multiple shares to maximize performance.

Table storage

Azure Table storage is a scalable NoSQL data store that enables you to store huge amounts of semistructured, nonrelational data. It does not allow you to do complex joins, use foreign keys, or execute stored procedures. Each table has a single clustered index that can be used to query the data quickly. You also can access the data by using LINQ queries and OData with the WCF Data Service .NET libraries. A common use of table storage is for diagnostics logging.

To use table storage, you have to create a storage account. Once you have a storage account, you can create tables and fill them with data.

A table stores entities (rows), each of which contains a set of key/value pairs. Each entity has three system properties: a partition key, a row key, and a timestamp. The partition key and row key combination must be unique; together they make up the primary key for the table. The PartitionKey property is used to shard (partition) the rows across different storage nodes, allowing for load balancing across storage nodes. All entities with the same PartitionKey are stored on the same storage node. The RowKey is used to provide uniqueness within a given partition.

To get the best performance, you should give a lot of thought to the PrimaryKey and RowKey and how you need to retrieve the data. The Azure Table service provides scalability targets for both storage account and partitions. The Timestamp property is maintained by Azure, and it represents the date and time the entity was last modified. Azure Table service uses this to support optimistic concurrency with ETags.

In addition to the system properties, each entity has a collection of key/value pairs called properties. There is no schema, so the key/value pairs can contain values of different properties. For example, you could be doing logging using the Semantic Logging Application Block, and one entity could contain a payload of {customer id, customer name, request date/time, request} and the next could have {customer id, order id, item count, date-time order filled}. You can store up to 252 key/value pairs in each table entity.

The number of tables is unlimited, up to the size limit of a storage account.

Tables can be managed by using the storage client library. The Table service also supports a REST API that implements the OData protocol; tables are addressable with the OData protocol using a URL in the following format:

[http://\[storage account name\]/table.core.windows.net/\[table name\]](http://[storage account name]/table.core.windows.net/[table name])

Queues

The Azure Queue service is used to store and retrieve messages. Queue messages can be up to 64 KB in size, and a queue can contain millions of messages, up to the maximum size of a storage account.

Queues generally are used to create a list of messages to be processed asynchronously. The Queue service supports best-effort first in, first out (FIFO) queues.

For example, you might have a worker role that continuously checks for messages on a queue. When it finds a message, it processes the message and then removes it from the queue. One of the most common examples is image processing.

Let's say you have a web application that lets a customer upload images into a container in blob storage. Your application needs to create thumbnails for each image. Rather than making the customer wait while this processing is done, you put a message on a queue with the customer ID and container name. Then you have a worker role that retrieves the message and parses it to get the customer ID and the container name. The worker role then retrieves each image, creates a thumbnail, and writes the thumbnail back to the same blob storage container as the original image. After all images are processed, the worker role removes the message from the queue.

What if you need to store information in the message that exceeds 64 KB? In that case, you could write a file with the information to a blob in blob storage and put the URL to the file in the queue message. The worker role could retrieve the message from the queue and then take the URL and read the file from blob storage and do the required processing.

One thing you need to be aware of is the Invisibility Timeout property on each message. When you retrieve a message from a queue, it is not deleted from the queue—you have to delete it when you're done with it. When the message is read from the queue, it becomes invisible. The Invisibility Timeout is the amount of time to allow for processing the message—if the message is not deleted from the queue within this amount of time, it becomes visible again for processing.

In general, you want to set this property to the largest amount of time that would be needed to process a message, so that while one instance of a worker role is processing it, another instance doesn't find it (visible) on the queue and try to process it at the same time. Regardless, it is critical that all processing for the message be idempotent, which means the same outcome should occur regardless of how often the message is processed, even if the processing is simultaneous.

You don't want to read the message from the queue, delete it from the queue, and then start processing it. If the worker role instance goes down, that queue entry will never be processed. Leaving the message on the queue (but invisible) until the processing has completed handles the case of the worker role instance shutting down—eventually the message will become visible again and will be processed by another instance of the worker role.

You can simulate a workflow by using a different queue for each step. A message can be processed from one queue from which it is deleted on completion, and then that processing can place a new message on a different queue to initiate processing for the next step in the workflow.

The Queue service provides poison message support through the dequeue count. The concern is that an invalid message could cause an application handling it to crash, causing the message to become visible on the queue again only to crash the next application to process it. Such a message is

referred to as a poison message. You can prevent this by checking the dequeue count for the message. If this exceeds some level, the processing of the message should be stopped, the message deleted from the queue, and a copy inserted in a separate poison message queue for offline review. You could process those entries periodically and send an email when an entry is placed on the queue, or you could just let them accumulate and check them manually.

If you want to process the queue messages in batches, you can retrieve up to 32 messages in one call and then process them individually.

Redundancy

What happens if the storage node on which your blobs are stored fails? What happens if the rack holding the storage node fails? Fortunately, Azure supports something called *replication*. There are four kinds of replication; you specify which one to use when you create the storage account. You can change the replication settings after they are set up, except in the case of Zone Redundant Storage.

- **Locally Redundant Storage (LRS)** Azure Storage provides high availability by ensuring that three copies of all data are made synchronously before a write is deemed successful. These copies are stored in a single facility in a single region. The replicas reside in separate fault domains and upgrade domains. This means the data is available even if a storage node holding your data fails or is taken offline to be updated.

When you make a request to update storage, Azure sends the request to all three replicas and waits for successful responses for all of them before responding to you. This means that the copies in the primary datacenter are always in sync.

If there is a problem with the entire datacenter going offline, your data will not be available, so it is recommended that you not use this for production data for which constant access is required.

- **Geo-Redundant Storage (GRS)** GRS makes three synchronous LRS copies of the data in the primary region for high availability, and then it asynchronously makes three LRS replicas in a paired region for disaster recovery. Each Azure region has a defined paired region within the same geopolitical boundary for GRS. For example, West US is paired with East US. This has a small impact on scalability targets for the storage account. The GRS copies in the paired region are not accessible to you, and GRS is best viewed as disaster recovery for Microsoft rather than for you. In the event of a major failure in the primary region, Microsoft would make the GRS replicas available, but this has never happened to date.
- **Read-Access Geo-Redundant Storage (RA-GRS)** This is GRS plus the ability to read the data in the secondary region, which makes it suitable for customer disaster recovery. If there is a problem with the primary region, you can change your application to have read-only access to the paired region. Your customers might not be able to perform updates, but at least the data is

still available for viewing, reporting, etc.

You also can use this if you have an application in which only a few users can write to the data but many people read the data. You can point your application that writes the data to the primary region but have the people only reading the data access the paired region. This is a good way to spread out the performance when accessing a storage account.

- **Zone Redundant Storage (ZRS)** This is a new option that applies only to block blobs. It replicates your data across two to three facilities, either within a single region or across two regions. This provides higher durability than LRS, but ZRS accounts do not have metrics or logging capability.

Creating and managing storage

In this section, we create a storage account and look at the options available. We also create a container in blob storage and show how to upload some blobs into the container by using Visual Studio. Then we create a table by using the Server Explorer in Visual Studio and show how to add some records to the table.

Create a storage account

To create a storage account, log into the Azure Preview Portal (portal.azure.com). Click +NEW in the lower-left corner and select Storage. You will see a screen similar to Figure 4-1.

Storage account

STORAGE ⓘ

azureessentials 😊

core.windows.net

PRICING TIER >

Standard-GRS

RESOURCE GROUP >

Default-Storage-WestUS

SUBSCRIPTION >

Azure Free Trial

LOCATION >

West US

DIAGNOSTICS >

Not configured

Add to Startboard

Create

FIGURE 4-1 Add new storage account.

First you need to fill in a name for the storage account. This will be the storage account name used in the endpoints for blobs, files, tables, and queues. In Figure 4-1, the storage account name is `azureessentials`. This means the blobs (for example) will be addressable as `http://azureessentials.blob.core.windows.net`.

The next field is the PRICING TIER. If you click this, you should see something similar to Figure 4-2.



FIGURE 4-2 Select a pricing tier.

This is the redundancy mentioned earlier in this chapter. G1 is Geo-Redundant Storage (GRS), L1 is Locally Redundant Storage (LRS), and R1 is Read-Access Geo-Redundant Storage (RA-GRS).

If you click BROWSE ALL PRICING TIERS at the bottom of this window, it will show more information about each tier and show Zone Redundant Storage (ZRA) as an option. The detail screen also shows the cost per 100 GB per month.

Select Locally Redundant Storage (LRS) because this is for testing and development, not for production, and click Select at the bottom of the screen.

The next field is RESOURCE GROUP; this is used to manage a collection of assets or resources. For example, if you were only going to use this storage account with a specific website, you would put them both in the same resource group. This allows you to manage those resources together. We're not going to do that at this time. If you click RESOURCE GROUP, you have the option to create a new group or select one that has already been set up for you (see Figure 4-3).

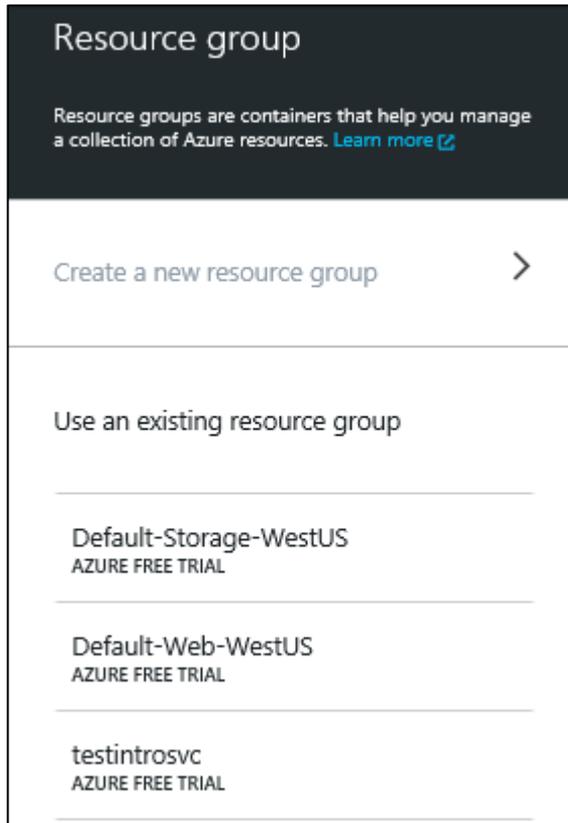


FIGURE 4-3 Select a resource group.

In this subscription, there have been websites, storage accounts, and cloud services created in the Azure Management Portal, so there are resource groups for those. We don't want this storage account to be associated with any other resources, so select Create A New Resource Group, type in a name, and click OK at the bottom of the Create Resource Group blade.

Next, the subscription of the current account logged in to the portal is displayed. If you are the administrator on more than one subscription with the same Microsoft account, clicking SUBSCRIPTION will show the other accounts, and you can select the account to which you want this storage account tied.

Next is LOCATION. This is the region in which the storage account will be located. Select the one

closest to your application so you will have the minimum latency.

Next is DIAGNOSTICS. If you click that, it will show the STATUS on the next screen. This is the toggle for collecting analytics on the storage account. You will be billed for storage transactions when writing the information to the storage account. Turn on the diagnostics and click OK at the bottom of the blade.

On the Add Storage Account screen, select the Add To Startboard check box and click Create at the bottom of the screen. Azure will provision your storage account and add it to your Startboard for quick access.

If you click your new storage account from the Startboard, a blade will be displayed with all the information about your storage account, the top of which should be similar to Figure 4-4.

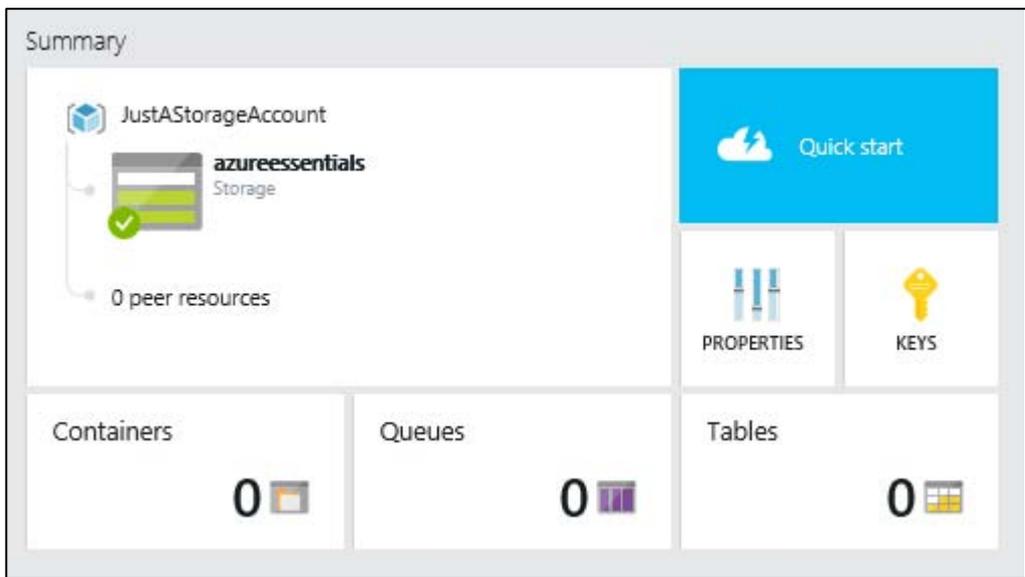


FIGURE 4-4 Displaying the new storage account.

You can click **PROPERTIES** to see the information about your storage account, including the endpoints for the Blob service, Queue service, and Table service. You can click **KEYS** to retrieve the access keys or regenerate them in case you need to change them. One of the keys plus the storage account name will be used to access the storage account.

Other sections in this blade include Monitoring and Usage. You must have Diagnostics turned on to use these features. There is a set of default metrics that Azure will retain. You also can select additional metrics and set the retention time. You don't pay for ingress with Azure, but you do pay for egress, so the default metrics include the total egress for blobs, tables, and queues. You also can create alerts for any of the metrics so you will receive notifications if any of your alert rules are triggered.

If you want to set up a custom domain, this feature is not yet available in the Azure Preview Portal.

However, there is a button to take you to the Azure Management Portal, or you can log in directly to *manage.windowsazure.com*.

Create a container and add blobs

Now that we have a new storage account, let's create a container in our new storage account and upload some files to it. Log into the Azure Preview Portal (*portal.azure.com*) and click the new storage account that you added to your Startboard. Next click CONTAINERS in the SUMMARY section (displayed previously in Figure 4-4). You should see something similar to Figure 4-5.

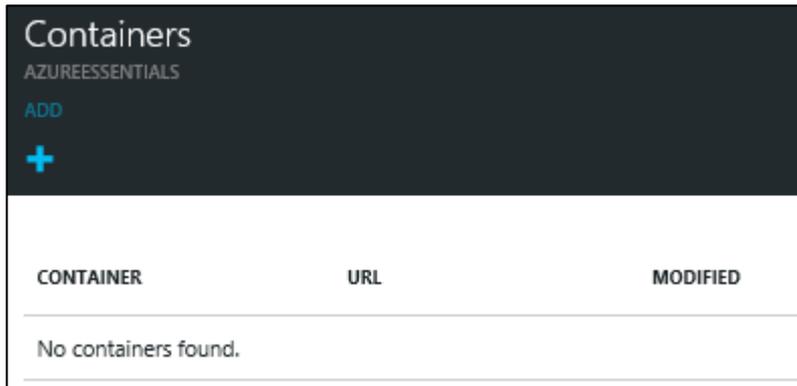


FIGURE 4-5 Containers blade.

Because the storage account was just created, there are no containers. Click ADD+. You will be prompted for the container name and the access type, as shown in Figure 4-6.

Add a container

NAME

Access type

Private
No anonymous access

Blob
Access blobs via anonymous requests

Container
List and access blobs via anonymous requests

FIGURE 4-6 Add a container.

The Access Type defines who can access the blobs and the container. If you set this to Private, the container and the blobs in the container can only be accessed by someone who has the account credentials (account name and key) or a URL that includes a security token. If you set this to Blob, then anyone with a URL can view the associated blob but cannot view the container properties and metadata or the list of blobs in the container. If you set this to Container, then everyone has read access to the container and the blobs therein.

Select Blob and click OK at the bottom of the screen to add the container.

Note that the URL includes the container name. This, concatenated with the blob name, will be the URL to each of the blobs in the container. Now let's add some blobs. Click the container, and you will see a screen that says This Container Has No Blobs.

There are many ways to upload blobs into your container. For a listing of various Azure storage explorers, check out this blob entry by the Azure Storage Team:
<http://blogs.msdn.com/b/windowsazurestorage/archive/2010/04/17/windows-azure-storage-explorers>.

[aspx](#).

You also can also upload blobs by using the Server Explorer in Visual Studio. This requires the storage account name and key. As noted previously, you can log into the Azure Preview Portal and click your storage account on the Startboard. Then, in the SUMMARY section, click KEYS and copy one of the keys to the Windows Clipboard.

In Visual Studio, open the Server Explorer. Select Windows Azure > Storage. You might be prompted for your Microsoft account credentials; if so, fill them in. In the Server Explorer > WindowsAzure section, right-click Storage and select Attach External Storage. You will be prompted for the storage account credentials (see Figure 4-7).

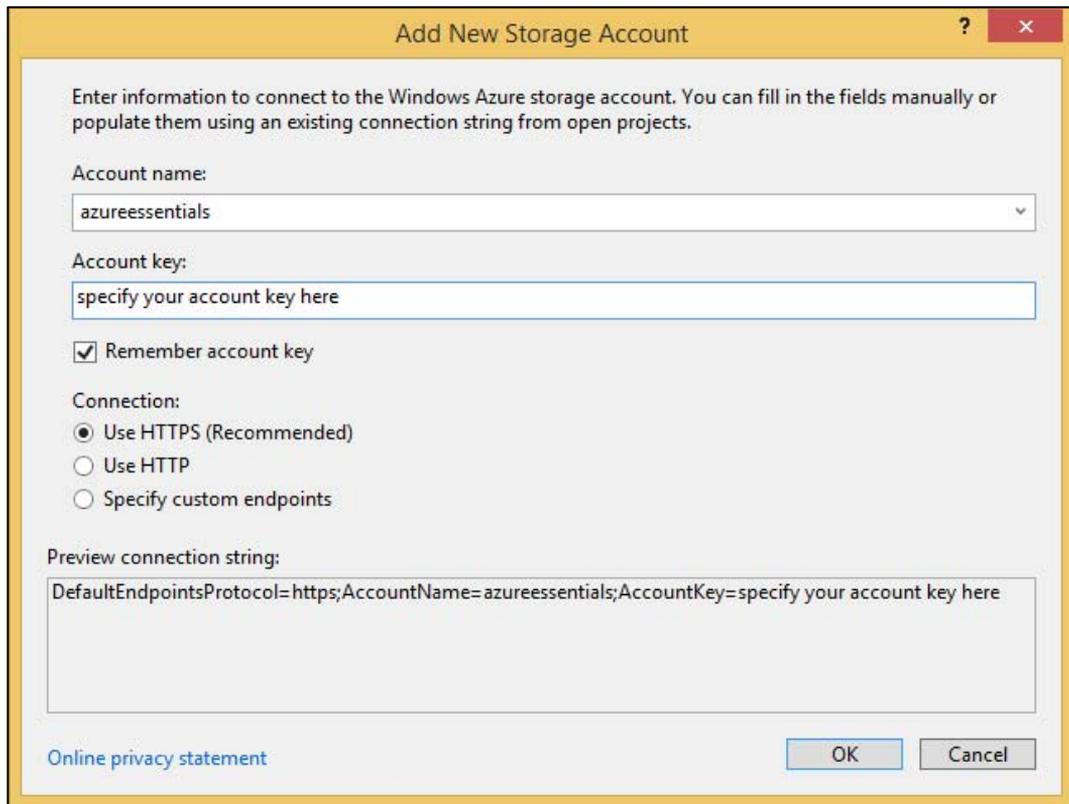


FIGURE 4-7 Attach external storage.

Enter the name of the storage account that you added in the previous section and paste the account key. Click OK to add the storage account. You now can see the storage account added to the Storage branch of Server Explorer, as shown in Figure 4-8.

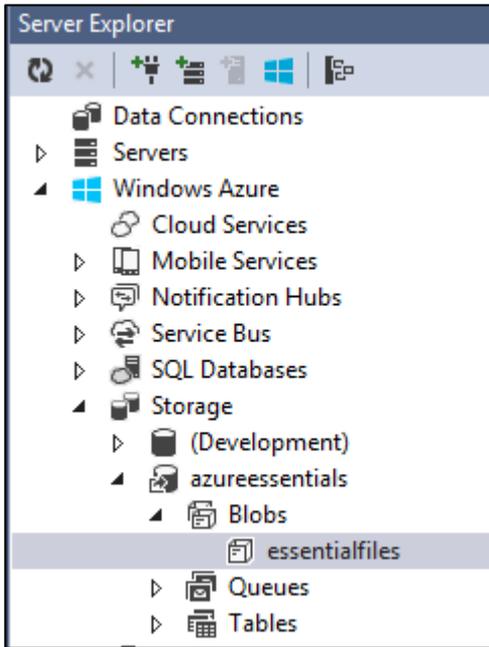


FIGURE 4-8 Display a new storage account.

You can see the essentialfiles container that was added earlier. Double-click the container; it will open the container in a separate window (see Figure 4-9).



FIGURE 4-9 Accessing the blobs in a container.

You now can upload blobs by clicking the icon with the up arrow. You will see a standard Windows File Explorer dialog box, and you can select one or more files to be uploaded and click OK. The Windows Azure Activity Log window will display and show if the upload was successful. After uploading several files, you should see something similar to Figure 4-10.

Name	Size	Last Modified (UTC)	Content Type	URL
Azure.jpg	78.5 KB	9/28/2014 8:50:54 PM	image/jpeg	https://azureessentials.blob.cor...
BluebellsAndBeechTrees.jpg	1.3 MB	9/28/2014 8:48:30 PM	image/jpeg	https://azureessentials.blob.cor...
Chewbacca.jpg	53.9 KB	9/28/2014 8:51:23 PM	image/jpeg	https://azureessentials.blob.cor...
Stormtroopers_On_Vacation.png	171.9 KB	9/28/2014 8:51:30 PM	image/png	https://azureessentials.blob.cor...
despicableme.jpg	190.5 KB	9/28/2014 8:51:41 PM	image/jpeg	https://azureessentials.blob.cor...
oreos.png	163.5 KB	9/28/2014 8:51:41 PM	image/png	https://azureessentials.blob.cor...
regrets.jpg	50.9 KB	9/28/2014 8:51:41 PM	image/jpeg	https://azureessentials.blob.cor...
shark.jpg	48.2 KB	9/28/2014 8:51:41 PM	image/jpeg	https://azureessentials.blob.cor...

FIGURE 4-10 View of the container after uploading files.

From Server Explorer, you also can delete blobs, open them, and download them. You even can add containers by right-clicking Blobs and selecting Create Blob Container.

This is a pretty simple implementation of accessing blob storage. It does not allow you to upload or download folders full of images. For more sophisticated applications, check out the list provided earlier in this section.

Create a table and add records

Let's create a table in our storage account and add some entities to it. You can use one of the storage explorer tools mentioned in the last section, but let's see how easy it is to use Visual Studio to do this task.

If you've done the steps in the last section to add blobs to blob storage, this will be just as easy. If you haven't, you want to open Visual Studio > Server Explorer and add the external storage account that you want to use.

In Server Explorer, right-click Tables and select Create Table. You will be prompted for the name of the table, which must be unique within your storage account. After clicking OK to create the new table, double-click the table name to see something similar to Figure 4-11.

essentialtbl [Table] - [X]

Enter a WCF Data Services filter to limit the entities returned

Table contains no entities. Start using the table by selecting the Add Entity button.

PartitionKey	RowKey	Timestamp

FIGURE 4-11 Editing the new table.

If you use this feature to open a table with many entities, you can filter the view by entering a WCF Data Services filter in the text box and clicking the green triangle to apply it.

We don't have any entities, so let's add some by clicking the icon with the + in it.

As discussed in the section "Table storage" earlier in this chapter, you have to think about what you want to use for PartitionKey and RowKey to get the best performance.

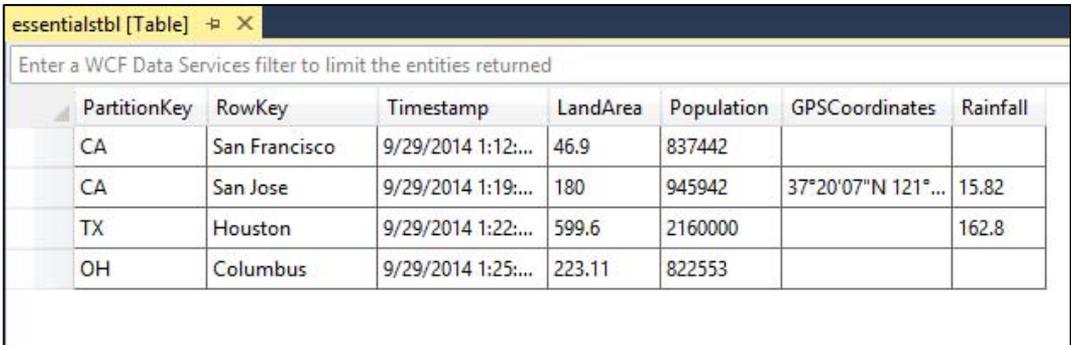
For this example, let's use state abbreviation for the PartitionKey and city name for the RowKey. For properties, add Population as Int32 and LandArea as a Double. Fill in values for each of the fields. Figure 4-12 shows what the entity looks like before adding it to the table.

	Name	Type	Value
	PartitionKey	String	CA
	RowKey	String	San Francisco
	Population	Int32	837442
X	LandArea	Double	46.9

FIGURE 4-12 Add an entity to the table.

Click OK to save the entity. Add another entity, and this time, add another property besides Population and LandArea, such as GPSCoordinates. Add a couple more entities, including whatever properties you want. If you want to edit an entity after saving it, you can right-click the entity and select Edit. You also can delete entities using this view.

After entering a few entities, you should have something similar to Figure 4-13.



The screenshot shows a window titled 'essentialstbl [Table]' with a search filter 'Enter a WCF Data Services filter to limit the entities returned'. Below the filter is a table with the following data:

PartitionKey	RowKey	Timestamp	LandArea	Population	GPSCoordinates	Rainfall
CA	San Francisco	9/29/2014 1:12:...	46.9	837442		
CA	San Jose	9/29/2014 1:19:...	180	945942	37°20'07"N 121°...	15.82
TX	Houston	9/29/2014 1:22:...	599.6	2160000		162.8
OH	Columbus	9/29/2014 1:25:...	223.11	822553		

FIGURE 4-13 View the table after adding entities.

You can see the PartitionKey and RowKey combination is unique for all of the entities. The rest of each row in the table is the list of key/value pairs. Not all entities have the same properties. The entity for San Francisco only has LandArea and Population; the entity for San Jose is the only one with GPSCoordinates. This is a strength of Azure Tables—the key/value pairs can vary for each entity.

You can create tables by using a designer such as this one in Visual Studio, but for adding, changing, and deleting entities, you probably will write your own code using the storage client library. For examples, please check out this link:

<http://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-tables/>.

Chapter 5

Azure Virtual Networks

In this chapter, we take a look at Azure Virtual Networks. We discuss the various components and see how to create a virtual network in Azure. We also see how to set up and use a point-to-site network.

The demos in this chapter are performed using the Microsoft Azure Management Portal (manage.windowsazure.com), because these features have not yet been migrated to the Microsoft Azure Preview Portal.

What is a virtual network

Overview

Virtual networks (VNETs) are used in Azure to provide a layer of security and isolation to your services. VMs and services that are part of the same virtual network can access each other. By default, services outside the virtual network cannot connect to services within the virtual network. You can, however, configure the network to allow access to the external service.

Services that talk to each other within a virtual network do not travel through the Azure Load Balancer, which gives you better performance. It's not significant, but sometimes every little bit counts.

Here's an example of when you might want to use a virtual network. Let's say you have a front-end web application running in a cloud service using a back-end database running in a virtual machine. You can put the back-end database in the same virtual network as the cloud service; the web application will access the database over the virtual network. This allows you to use the back-end database from the cloud service without the database being accessible on the public Internet.

You can add a Virtual Network Gateway to a virtual network and use it to connect your on-premises network to Azure, effectively making the virtual network in Azure an extension of your on-premises network. This provides the ability to deploy hybrid cloud applications that securely connect to your on-premises datacenter. The Virtual Network Gateway is a fully managed service in Azure.

More complex features available include multisite VPNs, in-region VNet-to-VNet, and cross-region VNet-to-VNet. Most cross-premises connections involve using a VPN device to create a secure connection to your virtual network in Azure. VNet-to-VNet connectivity uses the Azure Virtual Network Gateway to connect two or more virtual networks with IPsec/IKE S2S VPN tunnels. This gives you flexibility when connecting one or more on-premises sites with your virtual networks. For example, this gives you the ability to have cross-region geo-redundancy, such as SQL Always On across different Azure regions.

Definitions

When creating a virtual network, there are a few things you need to know, such as the address space, subnets, and DNS servers that you want to use.

Virtual network address spaces

When you set up a virtual network, you specify the topology of the virtual network, including the available address spaces and subnets. If the virtual network is to be connected to other virtual networks, you should select address ranges that are not overlapping. This is the range of addresses that the VMs and services in your network can use. Because these are private and cannot be accessed from the public Internet, you must use unroutable IP addresses, specified in CIDR notation, such as 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16. Unroutable IP addresses will not accept traffic on the public Internet—they can be used only on an internal network.

CIDR specifies an address range using a combination of an IP address and its associated network mask. CIDR notation uses this format: `xxx.xxx.xxx.xxx/n`, where `n` is the number of leftmost '1' bits in the mask. For example, 192.168.12.0/23 applies the network mask 255.255.254.0 to the 192.168 network, starting at 192.168.12.0. This notation therefore represents the address range 192.168.12.0–192.168.13.255. Fortunately, the Azure Management Portal displays this information in a drop-down list so you don't have to do bit-wise math to figure it out!

10.0.0.0/8 gives you a usable address range of 10.0.0.0–10.255.255.255. You can certainly use this, but if 10.0.0.0 is being used elsewhere in the network, either on-premises or within Azure, you want to be sure you have no overlap. One way to do this is to specify an address space that is smaller but still has the capacity to hold everything you want to put in it. For example, if you are just going to put a handful of VMs in your virtual network, you could use 10.0.0.0/27, which gives you a usable address range of 10.0.0.0–10.0.0.31.

If you work within an organization in which someone else is responsible for the internal networks, you should confer with that person before selecting your address space to make sure there is no overlap and to let him or her know what space you want to use so he or she doesn't try to use the same range of IP addresses.

Subnets

After specifying your virtual network address space(s), you can create one or more subnets for your virtual network. You do this to break up your network into more manageable sections. For example, you might assign 10.1.0.0 to VMs, 10.2.0.0 to back-end cloud services, and 10.3.0.0 to SQL Server VMs. Note that Azure reserves the first four addresses in each subnet for its own use.

By default, there is no security boundary between subnets, so services in each of these subnets can talk to one another. However, Microsoft recently announced the ability to set up Network Security Groups, which are associated with sets of allow and deny rules for the Internet, the virtual network, or IP ranges (expressed in CIDR form). Each Network Security Group then can be associated with a subnet

or VM, allowing the creation of subnet ACLs and VM ACLs so that you control the network traffic allowed to and from subnets and VMs.

Using subnets separates traffic by broadcast domain and requires routing to pass traffic to another subnet. This is useful in preventing chatty traffic from affecting other traffic on the network. For example, many organizations run their VoIP traffic on a different subnet from their workstations, keeping things like large downloads from affecting voice traffic.

DNS servers

If you want to refer to your VMs or role instances by host name or fully qualified domain name (FQDN) directly, rather than using an IP address and port number, you need a DNS service to provide name resolution. It is helpful to figure this out before deploying VMs or role instances. You can change it later, but if you do you will have to reboot all of the VMs in the virtual network because the DNS server information is injected into the settings at startup.

There are two options: you can use the Azure-provided name resolution or you can specify a DNS server that is not maintained by Azure, such as one that is used by your on-premises infrastructure or one that you set up and maintain in an Azure VM.

If you need name resolution across cloud services, you need to use your own DNS server. For example, if you have two VMs located on the same virtual network and you want them to be able to communicate by host name, you will need your own DNS server solution. If you don't have any on-premises DNS servers that you want to use, you can add a VM running DNS to your virtual network.

For more information, please take a look at this MSDN article on Name Resolution (DNS): <http://msdn.microsoft.com/en-us/library/azure/jj156088.aspx>.

Creating a virtual network

To put VMs into a virtual network, you create the virtual network and then, as you create each VM, you assign it to the virtual network and subnet. VMs and cloud services acquire their network settings during deployment.

VMs are assigned an IP address when they are deployed. If you deploy multiple VMs into a virtual network or subnet, they are assigned IP addresses as they boot up. A DIP is the internal IP address associated with a VM. You can allocate a static DIP to a VM or role instance. If you do this, you should consider using a specific subnet for static DIPs to avoid accidentally reusing a static DIP for another VM.

If you create a VM and later want to migrate it into a virtual network, it is not a simple configuration change. You have to redeploy the VMs into the virtual network. The easiest way to do this is to delete the VM, but not any disks attached to it, and then re-create the VM using the original disks in the virtual network.

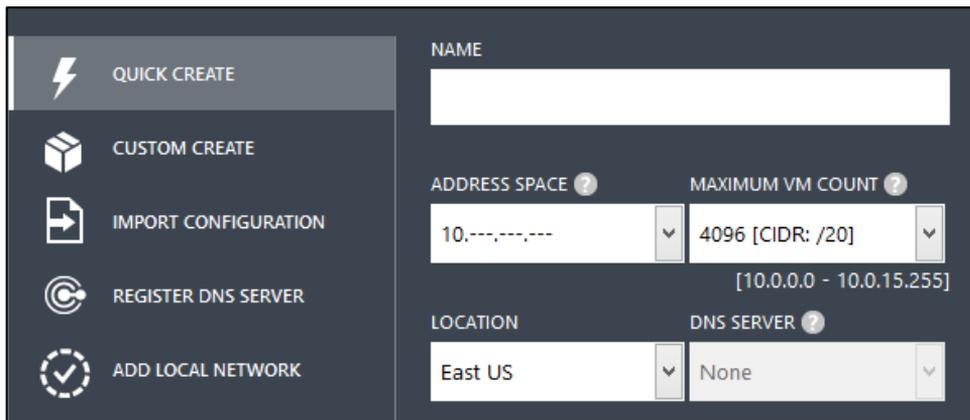
To put a cloud service (web roles and worker roles) into a virtual network, you just need to add a

NetworkConfiguration Section to the Service Configuration file that shows the virtual network information. When deployed, the cloud service will be placed in the virtual network.

Quick Create

Let's try the Quick Create option for creating a virtual network. Log into the Azure Management Portal (manage.windowsazure.com).

1. Click the +NEW button at the bottom of the screen and then select NETWORK SERVICES > VIRTUAL NETWORK > QUICK CREATE. You should see something similar to this:



The screenshot shows the 'QUICK CREATE' interface for a virtual network. On the left is a sidebar with five options: 'QUICK CREATE' (selected), 'CUSTOM CREATE', 'IMPORT CONFIGURATION', 'REGISTER DNS SERVER', and 'ADD LOCAL NETWORK'. The main area contains the following fields:

- NAME:** An empty text input field.
- ADDRESS SPACE:** A dropdown menu with the value '10.---.---.---' selected.
- MAXIMUM VM COUNT:** A dropdown menu with the value '4096 [CIDR: /20]' selected. Below it, the range '[10.0.0.0 - 10.0.15.255]' is displayed.
- LOCATION:** A dropdown menu with the value 'East US' selected.
- DNS SERVER:** A dropdown menu with the value 'None' selected.

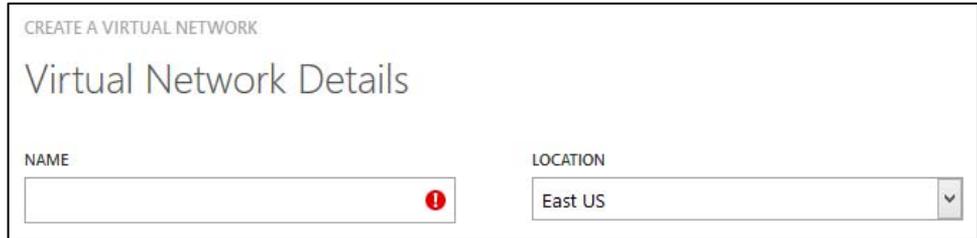
2. Fill in the NAME you want your network to have.
3. Next, select your ADDRESS SPACE. As discussed previously, you will find the drop-down list has only three entries from which to choose: 10.0.0.0, 172.16.0.0, and 192.168.0.0. Select 10.---.---.---.
4. The MAXIMUM VM COUNT is the count of the range of IP addresses included for the selected CIDR. If you look at the values in the drop-down list, you will see the highest CIDR (and thus the fewest number of IP addresses) is 20, which gives you 4,096 addresses.
5. For LOCATION, select the desired region.
6. For DNS SERVER, if you set this to None, the name resolution will be provided by Azure. If you want to have name resolution between this virtual network and your on-premises network, you should specify the DNS servers you are using for your on-premises name resolution. For this example, just leave it set to None.
7. Click the CREATE A VIRTUAL NETWORK check mark at the bottom of the screen. Azure will now provision your virtual network.

When it's finished, you will have a virtual network into which you can deploy your VMs and services.

Custom Create

Now let's look at the options available when doing a Custom Create of a virtual network.

1. In the Azure Management Portal, select **+NEW > NETWORK SERVICES > VIRTUAL NETWORK > CUSTOM CREATE**. You should see something similar to this:



2. Fill in a unique NAME and set the LOCATION to the desired region, then click the arrow on the lower-right corner of the screen. On the next page, leave the DNS SERVERS fields blank to use name resolution provided by Azure.



3. This is the point at which you would specify that you want to configure a point-to-site and/or site-to-site VPN. We'll see how to set up a point-to-site network later in this chapter. For now, leave these check boxes unselected and click the right arrow at the bottom of the screen to continue to the next page, shown here:

CREATE A VIRTUAL NETWORK

Virtual Network Address Spaces

ADDRESS SPACE	STARTING IP	CIDR (ADDRESS COUNT)	USABLE ADDRESS RANGE
10.0.0.0/8	10.0.0.0	/8 (16777...	10.0.0.0 - 10.255.255.255
SUBNETS			
Subnet-1	10.0.0.0	/11 (2097...	10.0.0.0 - 10.31.255.255

add subnet

add address space

- This is where you can specify the address space(s) and subnets for each. If you look at the values in the drop-down list for STARTING IP, you will find the usual suspects: 10.0.0.0, 172.16.0.0, and 192.168.0.0. Select 10.0.0.0.
- Next, you can specify the CIDR. Select /20, which provides an address range of 10.0.0.0 to 10.0.15.255.
- For subnets, let's set up two: one for 10.0.0.0 through 10.0.0.255 and one for 10.0.1.0 through 10.0.1.255. Click Subnet-1 and change this name to something more meaningful. Leave the STARTING IP at 10.0.0.0, but change the CIDR to 24. This will give you the range 10.0.0.0 to 10.0.0.255.
- Next, click Add Subnet; this will add a row that you can edit. Set the name, set the starting IP address to 10.0.1.0, and set the CIDR to /24. This will give you the range 10.0.1.0 to 10.0.1.255.

You can add more address spaces, but we don't need to do that. After filling in the fields, you should have something similar to the screen shown here:

CREATE A VIRTUAL NETWORK

Virtual Network Address Spaces

ADDRESS SPACE	STARTING IP	CIDR (ADDRESS COUNT)	USABLE ADDRESS RANGE
10.0.0.0/20	10.0.0.0	/20 (4096)	10.0.0.0 - 10.0.15.255
SUBNETS			
MyVMs	10.0.0.0	/24 (256)	10.0.0.0 - 10.0.0.255
MyDataSvcs	10.0.1.0	/24 (256)	10.0.1.0 - 10.0.1.255

8. Click the check mark in the lower-right corner of the screen to create the network.

Now you have a virtual network with two subnets. In the case above, we would deploy the VMs and PaaS instances into the MyVMs subnet and data services (such as SQL Server VMs) into MyDataSvcs.

Using a network configuration file

Another way to configure a virtual network is to upload a network configuration file. It's difficult to create one of these from scratch; the easiest way is to download the current network configuration file, modify it, and then upload it again. The network configuration file applies to the entire subscription. One use case of this feature is if you have a subscription with virtual networks in it and you want to clone one or more of them to another subscription.

Export the network configuration file

Let's export the network configuration file. This exports the file for the entire Azure subscription to an XML file.

1. Go to the Azure Management Portal (*manage.windowsazure.com*) and click NETWORKS in the left column. This will show the networks you have set up.
2. Click EXPORT at the bottom of the screen. You will be prompted for subscription. If you are the admin on multiple subscriptions, it will show a drop-down list. Select the subscription for which you would like to export the network configuration file and click the check mark at the bottom of the screen.

For the two networks set up in the examples in this chapter, this is what the network configuration file looks like:

LISTING 5-1 Network Configuration File example.

```
<NetworkConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration">
  <VirtualNetworkConfiguration>
    <Dns />
    <VirtualNetworkSites>
      <VirtualNetworkSite name="contosonetWORK" Location="West US">
        <AddressSpace>
          <AddressPrefix>172.16.0.0/20</AddressPrefix>
        </AddressSpace>
        <Subnets>
          <Subnet name="Subnet-1">
            <AddressPrefix>172.16.0.0/23</AddressPrefix>
          </Subnet>
        </Subnets>
      </VirtualNetworkSite>
      <VirtualNetworkSite name="contosonetWORK2" Location="West US">
        <AddressSpace>
          <AddressPrefix>10.0.0.0/20</AddressPrefix>
```

```

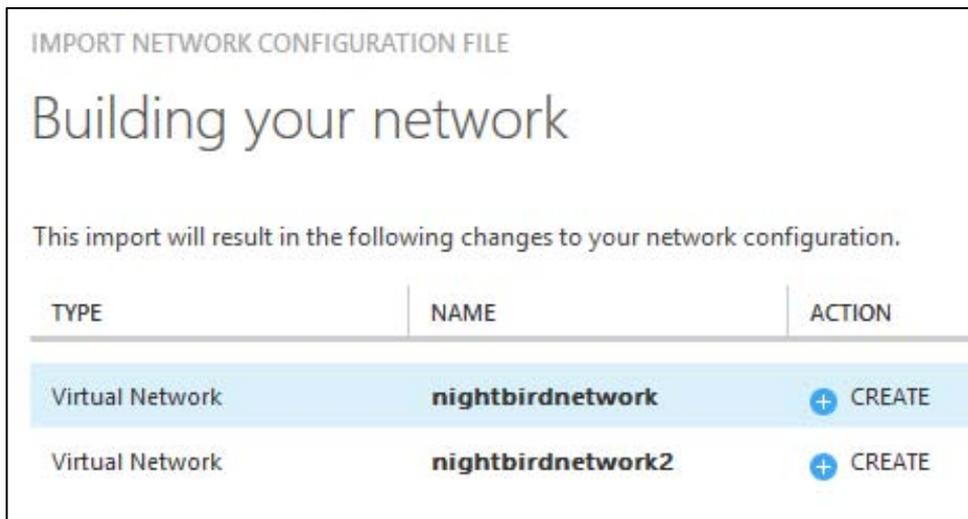
</AddressSpace>
<Subnets>
  <Subnet name="MyVMs">
    <AddressPrefix>10.0.0.0/24</AddressPrefix>
  </Subnet>
  <Subnet name="MyDataSvcs">
    <AddressPrefix>10.0.1.0/24</AddressPrefix>
  </Subnet>
</Subnets>
</VirtualNetworkSite>
</VirtualNetworkSites>
</VirtualNetworkConfiguration>
</NetworkConfiguration>

```

Import the network configuration file

Let's take this file and upload it to a subscription without any virtual networks. It should create an identical set of virtual networks. If you don't have another Azure subscription, delete the two virtual networks we created before importing this file so you can verify that new ones get created.

1. Log into the Azure Management Portal (*manage.windowsazure.com*). In the lower-left corner, click +NEW > Network Services > Virtual Network > Import Configuration.
2. You will be prompted to browse for the network configuration file. Browse for it and then click the arrow at the bottom of the screen. It shows what it is now going to do. Because I'm using a second subscription, it says it will create both of them.



Click the check mark at the bottom of the screen. Azure will now provision those two virtual networks with identical address spaces and subnets into the subscription.

What ifs

This raises a lot of questions because you have to download the whole configuration for the subscription and cannot download just the configuration for one of the virtual networks defined in the subscription:

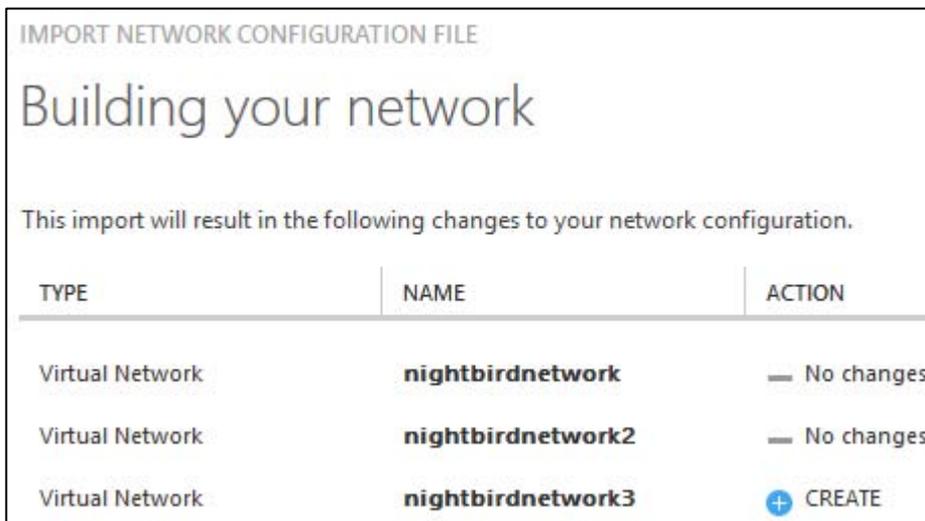
- Is there a way to modify the settings for one of the virtual networks?
- Can you add another virtual network?
- Can you remove one or more of the virtual networks?
- What if you already have VMs or services deployed into the virtual network?

Let's run through some scenarios and see how Azure handles them.

When you upload the network configuration file, Azure checks for differences. If any are found, it prompts you before committing the change.

What if you add a network?

If you add a network to the configuration file but don't change the other networks, you will see something like Figure 5-1, and you will have to approve the change before Azure will commit it.



TYPE	NAME	ACTION
Virtual Network	nightbirdnetwork	— No changes
Virtual Network	nightbirdnetwork2	— No changes
Virtual Network	nightbirdnetwork3	+ CREATE

FIGURE 5-1 Import dialog box after adding new network to network configuration.

What if you remove a network?

If you remove a network from the configuration, Azure will recognize that (Figure 5-2) and prompt you to approve the changes.

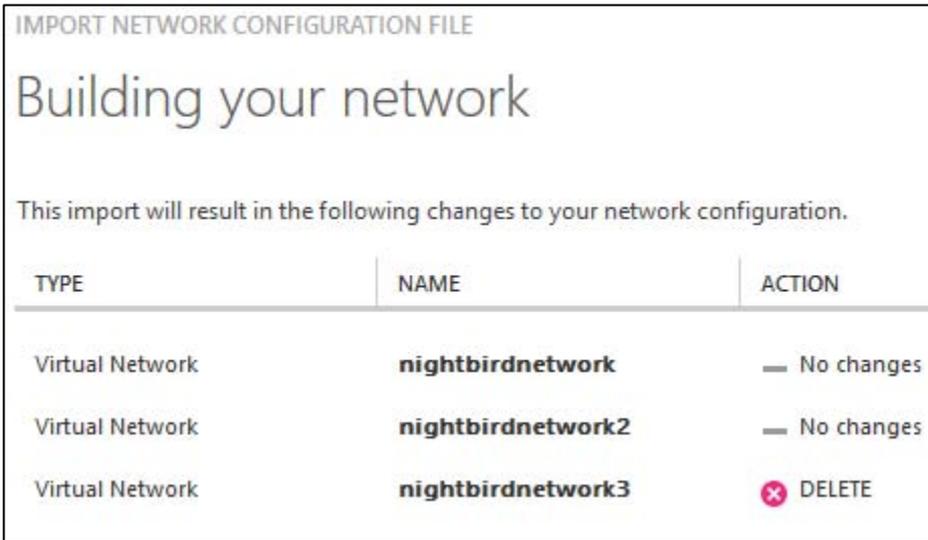


FIGURE 5-2 Import dialog box after removing network from network configuration.

If you try to remove a section that has VMs or other services deployed into it, the dialog box displayed in Figure 5-2 will be displayed, but when you try to commit the changes, you will get an error message saying that the network configuration file could not be imported.

What if you change a network?

If you change the configuration for a network that has no VMs or services deployed into it yet, Azure will show the dialog box displayed in Figure 5-3.

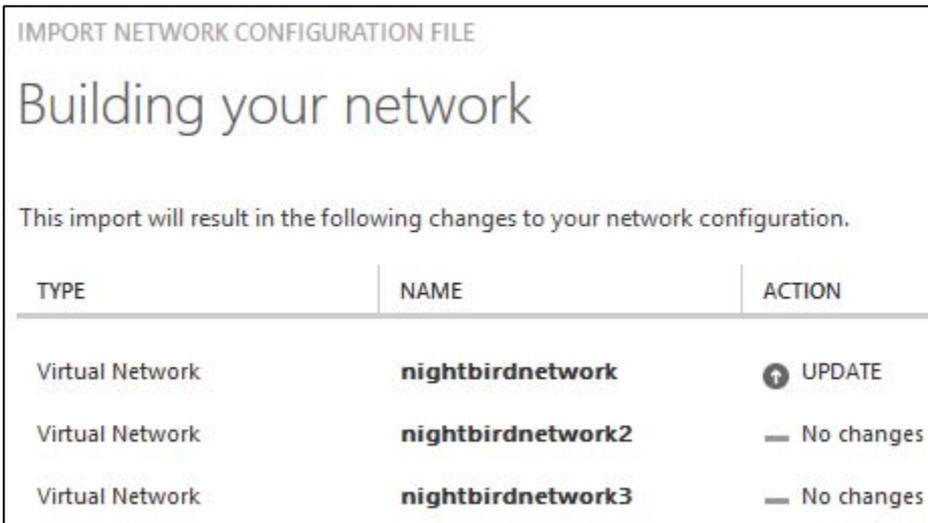


FIGURE 5-3 Import dialog box after changing one of the network's configurations (no VMs).

If you try to change a network that has VMs or other services deployed into it, you will still see this dialog box, but when you try to commit the changes, Azure will display an error message saying the network configuration file could not be imported.

In summary, if you upload a network configuration file that has changes to a network without anything deployed into it, the changes (whether additions, modifications, or removals) will be applied. If you upload a network configuration file that has changes to a network that has anything deployed into it (such as a cloud service or VM), your changes will be rejected.

Cross-premises connection options

There are many cases in which you might want to connect to your infrastructure in Azure from your on-premises network, a customer's site, your home network, or even a coffee shop, and you want to do this without compromising security. There are three options available in Azure to help you set up these cross-premises connections: site-to-site VPN, point-to-site VPN, and private VPN (ExpressRoute).

For both point-to-site and site-to-site connectivity, you must set up a VPN gateway in Azure. This gateway is the connection point into Azure from either the on-premises network (site-to-site) or the client machine (point-to-site). You'll see how to do this when we set up a point-to-site network later in this chapter.

Site-to-site connectivity

A site-to-site VPN lets you connect securely from your on-premises network to your virtual network in Azure. You have to have a public-facing IPv4 IP address and a compatible VPN device or Routing and Remote Access (RRAS) running on Windows Server 2012. For a list of valid devices and the configuration thereof, please refer to <http://msdn.microsoft.com/en-us/library/azure/jj156075.aspx>.

Once you have the connection up and running, resources on your local network such as computers and VMs can communicate with the resources in the virtual network on Azure. For example, if you host a company application on Azure, your employees can access and run that application securely using your site-to-site network.

You actually can use site-to-site connectivity to connect entire on-premises networks to virtual networks in Azure. A good example is a company that has multiple branch offices. You can establish a connection between each branch office's network and Azure.

Point-to-site connectivity

Point-to-site VPN enables you to connect from your local machine over a Secure Socket Tunneling Protocol (SSTP) tunnel to your virtual network in Azure. This uses certificate authentication between the client machine and the virtual network in Azure. This means you have to create some certificates and install them in the right places; we'll cover this in detail later in this chapter when we create a

point-to-site network.

It is recommended that you create a separate client certificate for each client that is going to access the point-to-site network and keep track of the certificate's thumbnail and on which machine it was installed. If you do this, and you later need to turn off access for one person, you can do that by invalidating the client certificate using the Azure subscription ID, the virtual network name, and the certificate thumbprint.

If you use the same client certificate on multiple machines, the only way to revoke access is to remove the root certificate in Azure, which revokes access for every client certificate that chains back to that root certificate.

You can connect up to 128 clients to the virtual network in Azure. (The maximum bandwidth is 80 MBPS per gateway.) The connection has to be configured on each client machine that you want to use. Once configured, the user can start the VPN by manually starting the connection, although you can configure the VPN to start automatically if needed.

We are going to cover how to set up a point-to-site network in depth later in this chapter.

Comparing site-to-site and point-to-site connectivity

There are several differences between these two forms of secure connections:

- You don't need a VPN device or RRAS to use a point-to-site network.
- With point-to-site, configuration must be done on each client machine. With site-to-site, no changes are required to the client machines.
- Point-to-site is a good choice when:
 - You only have a few clients that need to have access.
 - You don't have access to a VPN device that you can use for a site-to-site connection.
 - You want to connect securely when offsite (such as at a customer site or a coffee shop).
- You can have both point-to-site and site-to-site networks running simultaneously. If you can create a site-to-site network, you might use site-to-site for people on premises but allow point-to-site for people who need to connect from a remote location.

Private site-to-site connectivity (ExpressRoute)

Last, but not least, is private site-to-site connectivity, which in Azure means ExpressRoute. This is called private because the network traffic occurs over your network provider and does not go across the public Internet as it does with both site-to-site and point-to-site connectivity. This capability ensures that applications with privacy requirements can be developed and run on Azure. Using ExpressRoute also gives you increased reliability and speed and lower latency.

With this option, the network connects on your end to one of two scenarios: hardware colocated at an Exchange provider (such as Equinix or Level 3) or an additional site on your MPLS VPN-based WAN through a network server provider. The connection from the Exchange provider or the network service provider connects directly to Azure. A single ExpressRoute circuit can connect to multiple virtual networks in the same Azure geography (continent).

Inbound bandwidth to Azure is always free. Outbound data is unlimited if using ExpressRoute with a network service provider. With an Exchange provider, a significant amount of bandwidth is included. Because of this, if you have workloads in which huge amounts of data are leaving the Azure datacenter, using ExpressRoute can significantly lower the cost of that data transfer. Depending on the provider you select, the bandwidth can range from 10 MBPS to 10 GPBS.

This is the best solution for delivering enterprise-grade solutions. It is a good fit for applications or workloads that are mission critical to your company. The consistent network performance provided by ExpressRoute also makes it a good solution if you have SLAs in place with groups internal or external to your organization.

Point-to-site network

In this section, we see how to set up a point-to-site network and test it by deploying a VM into the network and connecting to it from the local machine.

Overview of setup process

Here are the steps we're going to follow to configure a point-to-site network to access from our local machine.

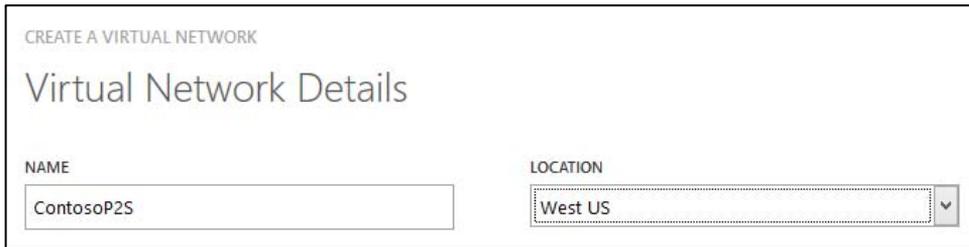
1. Create a virtual network with point-to-site connectivity enabled.
2. Deploy a VM into the virtual network.
3. Create a network gateway.
4. Create a self-signed root certificate.
5. Create a self-signed client certificate from the root certificate.
6. Export the client certificate from the certificate store.
7. Upload the root authentication certificate to Azure.
8. Install the client certificate on the client machine to authenticate to the virtual network.
9. Install the client VPN package.
10. Establish and verify the VPN connection.

Configuring point-to-site VPN

To start, log into the Azure Management Portal (*manage.windowsazure.com*). In this demo, we create the virtual network, deploy a VM into the network, and then configure the gateway for the point-to-site connectivity.

Create a virtual network

1. Click +NEW > NETWORK SERVICES > VIRTUAL NETWORK > CUSTOM CREATE. This opens the screen shown here:



The screenshot shows the 'CREATE A VIRTUAL NETWORK' interface. The main heading is 'Virtual Network Details'. There are two input fields: 'NAME' with the value 'ContosoP2S' and 'LOCATION' with a dropdown menu showing 'West US'.

2. Specify the NAME and select the desired LOCATION, then click the right arrow at the bottom of the screen.

This example will use ContosoP2S for NAME and West US for LOCATION.

3. On the DNS Servers And VPN Connectivity page, accept the defaults and select the Configure A Point-To-Site VPN check box to configure a point-to-site VPN.



The screenshot shows the 'CREATE A VIRTUAL NETWORK' interface for 'DNS Servers and VPN Connectivity'. It features a table for 'DNS SERVERS' with columns for 'ENTER NAME' and 'IP ADDRESS'. On the right, there are two sections: 'POINT-TO-SITE CONNECTIVITY' with a checked checkbox for 'Configure a point-to-site VPN', and 'SITE-TO-SITE CONNECTIVITY' with an unchecked checkbox for 'Configure a site-to-site VPN'.

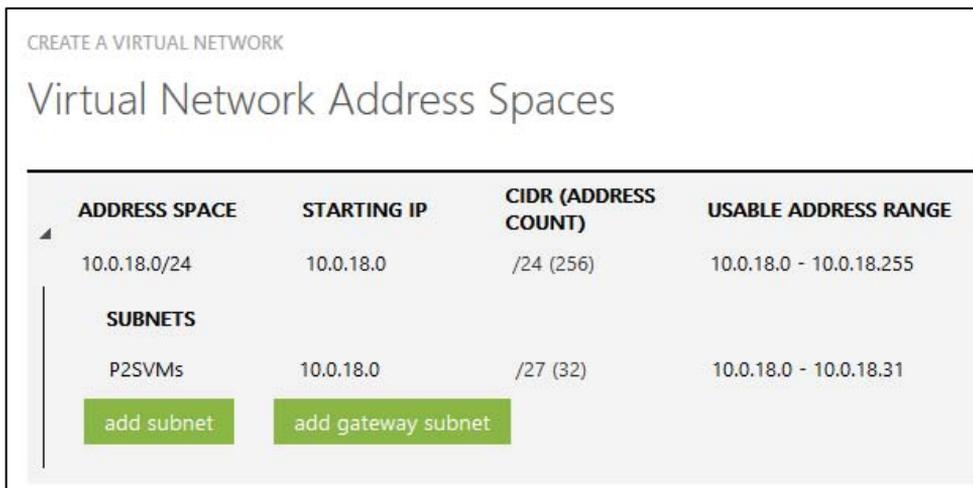
4. Click the right arrow to continue.
5. On the next screen, specify the IP address range from which your VPN clients will receive an IP address when connected. Let's use the default of 10.0.0.0/24.



6. Click the right arrow to continue.

Next, we set up the address space to be used by our virtual network:

1. Let's use starting IP 10.0.18.0 with a CIDR of /24. This gives us an address range of 10.0.18.0 through 10.0.18.255.
2. Rename the subnet to P2SVMs, with CIDR 27.

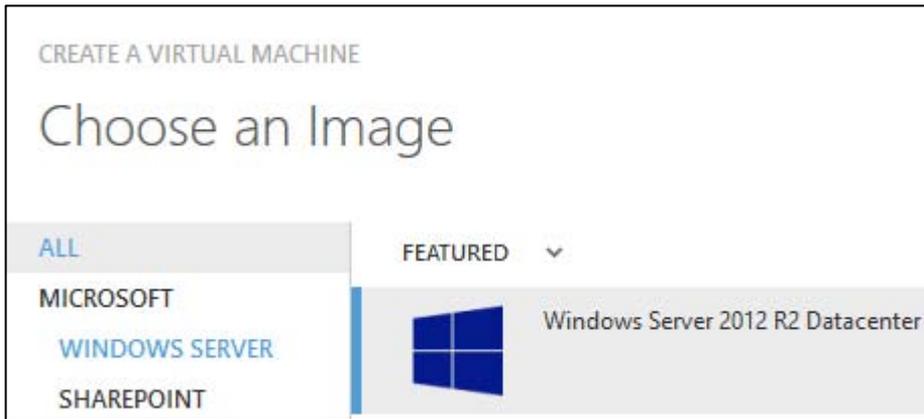


3. If we want to use a point-to-site network, we have to add a gateway subnet here. Click Add Gateway Subnet, which will add a default of 10.0.18.32/29. Accept the default and continue to the next screen by clicking the check mark on the lower-right side of the screen. At this point, Azure will create the virtual network, and then you can deploy a VM into the network.

Deploy a virtual machine into the virtual network

Because the rest of this chapter uses the Azure Management Portal, we're going to create our VM by using that portal. You also can create VMs by using the Azure Preview Portal, as illustrated in Chapter 3, "Azure Virtual Machines."

1. To deploy a VM into the virtual network, start by clicking +NEW > COMPUTE > VIRTUAL MACHINE > FROM GALLERY. Select Windows Server 2012 R2 Datacenter and click the right arrow on the bottom of the screen.



2. On the Virtual Machine Configuration screen, fill in the VM name and then specify the user name and password. This is the account used to log into the VM via Remote Desktop (RDP). Click the right arrow to continue to the next page.

CREATE A VIRTUAL MACHINE

Virtual machine configuration

VERSION RELEASE DATE ?

9/8/2014

VIRTUAL MACHINE NAME ?

contosovm

TIER

BASIC STANDARD

SIZE ?

A1 (1 core, 1.75 GB memory)

NEW USER NAME

contosouser

NEW PASSWORD ? CONFIRM

•••••••• ✓ ••••••••

3. On the second configuration screen, specify a unique cloud service DNS name. From the REGION/AFFINITY GROUP/VIRTUAL NETWORK drop-down list, select the virtual network you created earlier in this section. From the VIRTUAL NETWORK SUBNETS drop-down list, select the desired subnet. For STORAGE ACCOUNT, you can have Azure create a new one for you, or you can select one that you've already created in the same region. Click the right arrow on the bottom of the screen to continue.

CREATE A VIRTUAL MACHINE

Virtual machine configuration

CLOUD SERVICE ?

Create a new cloud service

CLOUD SERVICE DNS NAME

contosovmsvc .cloudapp.net

REGION/AFFINITY GROUP/VIRTUAL NETWORK ?

ContosoP2S

VIRTUAL NETWORK SUBNETS

P2SVMs(10.0.18.0/27)

STORAGE ACCOUNT

Use an automatically generated storage account

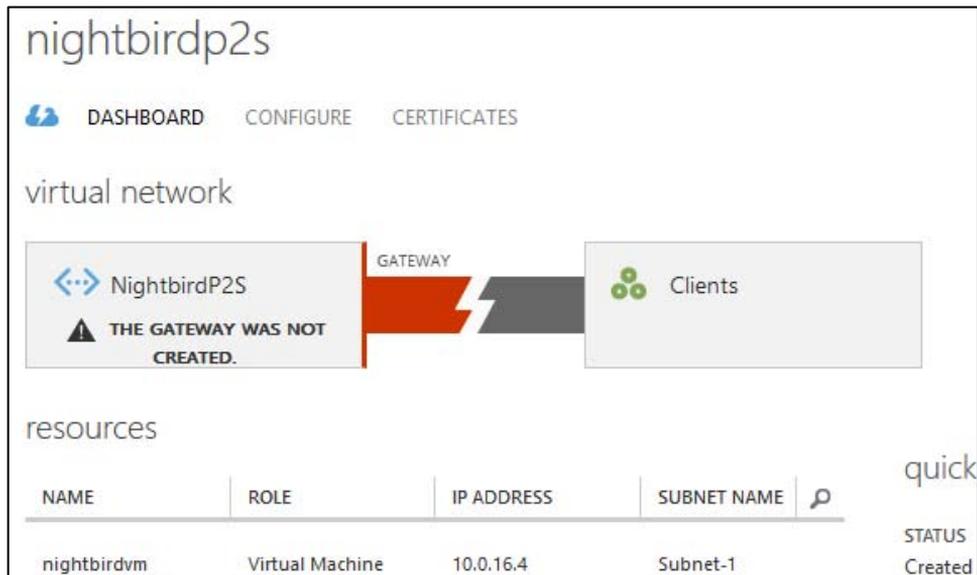
AVAILABILITY SET ?

(None)

4. On the last screen, just accept the defaults and click the check mark on the lower-right side of the screen. Azure will now provision and start your VM in your virtual network. While this is running, we can go back to our network and continue creating the network gateway.

Create the VPN gateway

1. Click the DASHBOARD tab at the top of the screen. It will tell you that the gateway hasn't been created yet.



2. Click the +GATEWAY button at the bottom of the screen. When asked if you want to create a gateway for your virtual network, click YES. Azure is now creating your gateway for your point-to-site connectivity. This is going to take a few minutes, so you can continue with the certificate tasks while this is working.

Create an authentication certificate for your virtual network

As discussed before, you need to create a self-signed root certificate and a client certificate for authentication when connecting to the VM on the network from the client. This is because rather than use password authentication, which is fairly weak, point-to-site connectivity uses certificate authentication. Someone without the correct client certificate installed will not be able to connect to the virtual network, even if he or she somehow obtains the IP address of the network.

Here are the steps we follow for generating the certificates.

1. Generate a self-signed root certificate.
2. Upload the root certificate to the Azure Management Portal.
3. Generate a client certificate that uses the root certificate you just created.
4. Export and install the client certificate on the client machine that is going to connect to the network.

To create certificates, you need the makecert.exe file. If you have any version of Visual Studio installed, you should have this. Some people can find this under C:\Program Files (x86)\Microsoft SDKs\Windows\v7.1A\Bin. If your machine is 32-bit, it might be under C:\Program Files\Microsoft SDKs\Windows\v7.1A\Bin. If you are running Windows 8.1, you might see it under C:\Program Files

(x86)\Windows Kits\8.1\Bin\x64 (or x86). The fastest way to find it is to open a command prompt window, go to the root of the C drive, and search for it by issuing this command:

```
dir/s makecert.exe
```

If you can't find it, install Visual Studio Express, which will include it in the installation.

Copy makecert.exe to a place to which you can easily navigate in the command window, such as C:\makecert\. Next, open a command window and navigate to that directory (cd C:\makecert). Now you're ready to create your certificates.

1. Create a self-signed root certificate. (Only self-signed root certificates are supported at the time of this writing.) This command will create a root certificate with the name ContosoP2SRoot stored in the current directory as ContosoP2SRoot.cer:

```
makecert -sky exchange -r -n "CN=ContosoP2SRoot" -pe -a sha1 -len 2048 -ss My  
.\ContosoP2SRoot.cer
```

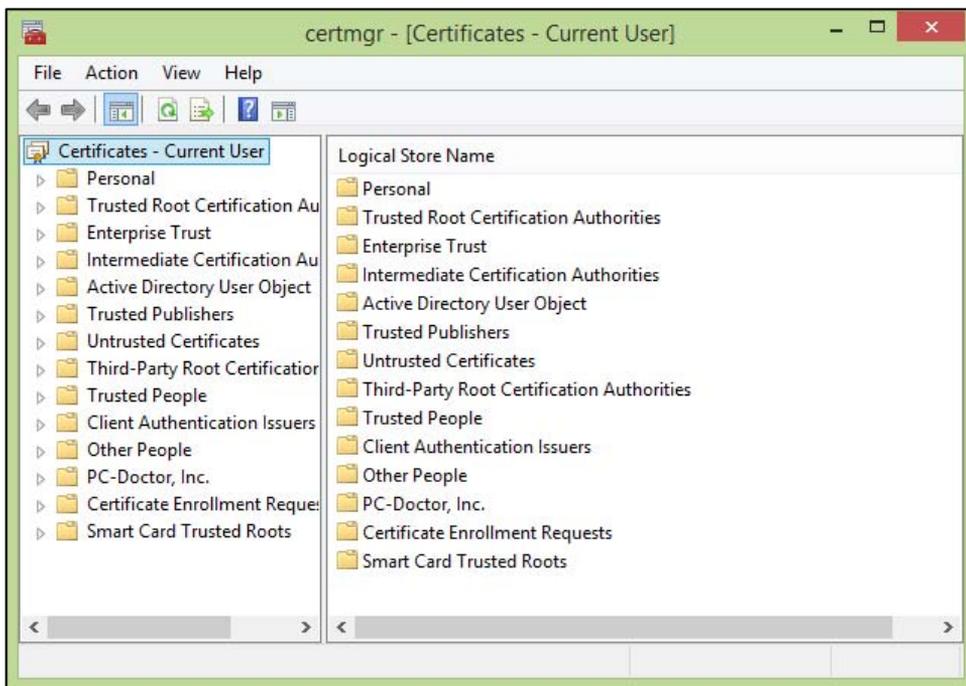
2. Create a self-signed client certificate using the root certificate you just created. The following command will create a client certificate with the name ContosoP2SClient:

```
makecert.exe -n "CN=ContosoP2SClient" -pe -sky exchange -m 96 -ss My -in  
"ContosoP2SRoot" -is my -a sha1
```

This creates and installs the client certificate on the local machine.

To use that certificate on other client machines, you will have to export the certificate, copy it to the other machine, and import it.

3. To export the client certificate to a file, open the certificate manager, find the certificate, and export it. To open the certificate manager, open the run box (WindowsKey+R), type **certmgr.msc**, and press Enter.



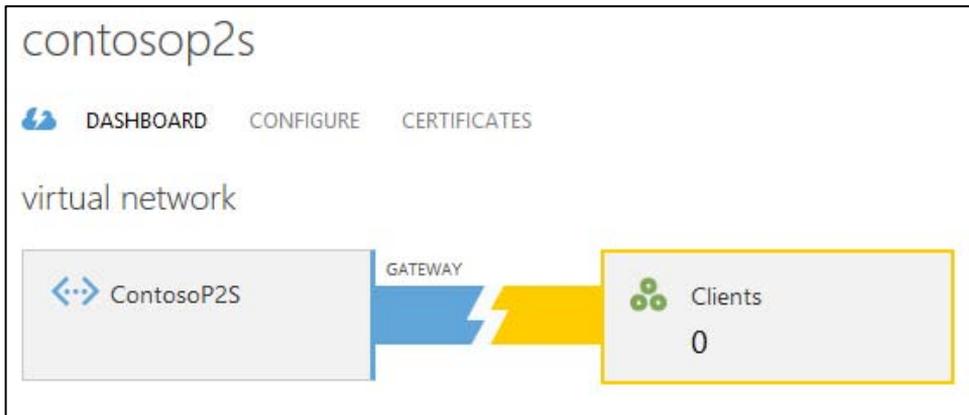
4. Open Personal, then Certificates. You will see a list of certificates on the right side. Find the one issued by ContosoP2SRoot that is named ContosoP2SClient.
5. Right-click the ContosoP2SClient certificate and select All Tasks > Export. This opens the Export dialog box. Click Next.
6. On the next screen, select Yes, Export The Private Key and click Next.
7. Accept the default selection of Personal Information Exchange—PKCS #12 (.PFX), with the Include All Certificates In The Certification Path If Possible check box selected, and click Next.
8. Click the Password check box, enter a password, and enter the confirmation password. Click Next.
9. Fill in a path and file name. This demo will use ContosoP2SClient.pfx. Click Save to accept the name and then click Next on the Certificate Export Wizard page. On the next page, click Finish. It will now export the client certificate to the location specified. Close the certificate manager and command window.

All client certificates that chain back to that root certificate will be valid when connecting a client machine to the point-to-site network. Microsoft recommends creating a separate client certificate for each client that is going to connect. If you do this and keep track of them, and then you need to revoke someone's access, you can invalidate that single client certificate.

Upload the self-signed root certificate

Next, you need to upload the root authentication certificate to Azure. This is used for the authentication handshake with the client machine.

1. Log into the Azure Management Portal (*manage.windowsazure.com*).
2. Click NETWORKS on the left side. Click the network you created earlier in this section.
3. Click CERTIFICATES at the top of the screen.
4. Click UPLOAD at the bottom of the screen. You want to find the first certificate you created, which was the root certificate. In this exercise, we created ContosoP2SRoot.cer. When you find the certificate on the local computer, select it and click the check mark to upload it.
5. Now if you go back to the DASHBOARD tab, you will see that the message saying you have no gateway and the message saying you need a certificate are both gone.



Install the client certificate (.pfx) to authenticate the virtual network

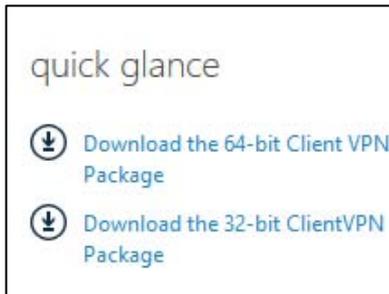
The client machine must have the certificate in its certificate store so the handshake with the virtual network will authenticate.

1. Navigate to the folder with the client certificate file (ContosoP2SClient.pfx in this demo).
2. Double-click the file to open it. This will launch the Certificate Import Wizard. Accept all defaults when stepping through the Certificate Import Wizard and enter the password when prompted. This is the password you set when you exported the certificate earlier.
3. When prompted to install the certificate, click Yes. You should see a dialog box indicating the import was successful.

Install the client VPN package

Download and install the VPN package on the client machine. This will add the virtual network to the client machine's network list.

1. When logged into the client computer, go to the Azure Management Portal and navigate to the DASHBOARD for your virtual network.
2. Under Quick Glance on the right side, there are two options to download the client VPN package.



3. Download the appropriate package for your machine (64-bit or 32-bit). When prompted, click Save instead of Run. This is going to have a name that looks a lot like a GUID, with the file extension .exe.
4. Because the file came from a location outside your computer, it might be blocked to help protect your computer, especially if you are running Windows 8 or higher. To check this and unblock it if needed, navigate to the downloaded file using Windows Explorer. Right-click it and select Properties. If it's blocked, click Unblock and then click OK.
5. Double-click the .exe file. When prompted to install the VPN client, click Yes.

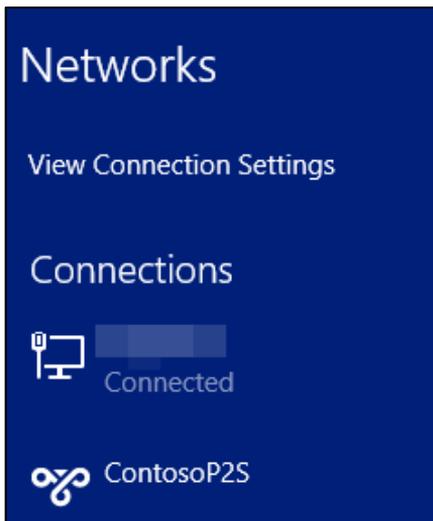
Connect to the virtual network through the VPN client

Let's connect from the client machine to the virtual network.

1. In the Azure Management Portal, click DASHBOARD for your virtual network.
2. In the Resources section, get the internal IP ADDRESS of the VM you created earlier in this section (highlighted here). You will need this when you RDP into the VM on the network.

resources			
NAME	ROLE	IP ADDRESS	SUBNET NAME 
contosovm	Virtual Machine	10.0.18.4	P2SVMs

- Click the Internet connection icon in the system tray (on the right side of the taskbar). From the list of connections displayed, select the virtual network. In this example, the name of the network is ContosoP2S.



- Click the virtual network and then click Connect.
- The following connection dialog box is displayed:



4. Click Connect. You will be prompted to elevate the privileges of Connection Manager.



5. Click Continue to elevate the privileges.

You are now connected to the VM through the virtual network, and you can RDP into the VM.

Connect to the VM using the internal IP address

Now that the network is up, let's connect to the VM we deployed into the virtual network by using

RDP.

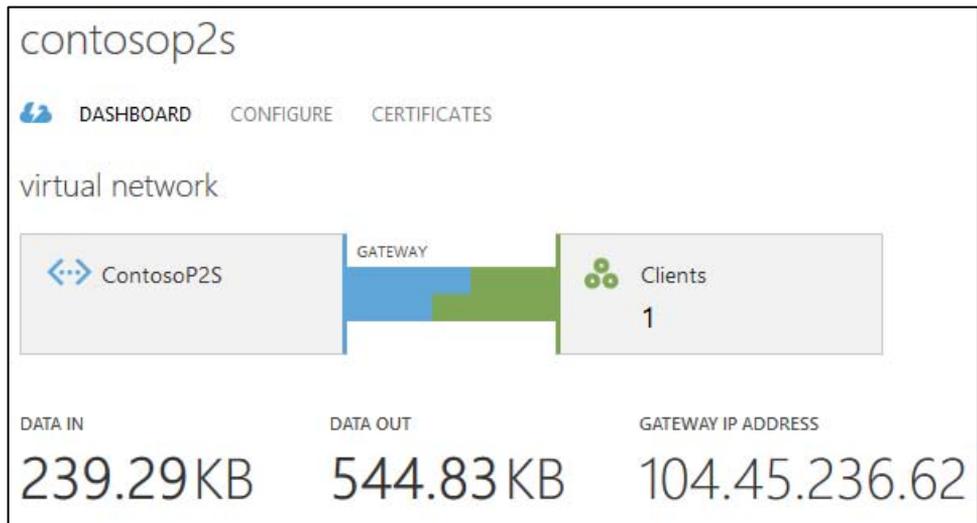
1. Run Remote Desktop. To do this, you can select it from the Start menu. You also can press WindowsKey+R, type **mstsc**, and press Enter.
2. In the Computer text box, type the internal IP address for the VM that you retrieved from the Azure Management Portal and then click Connect.



3. Log in with the credentials you set when you created the VM. Click Yes in the RDP warning dialog box.

This will log you into the VM across the virtual network. You can tell you are using the virtual network because you are connecting to the internal IP address, which is not publicly available.

4. If you go back to the DASHBOARD for the network, it will now show that it has one client connected (you might have to refresh the browser).



Test the VPN connection

To make sure that there is no public access to our VM, let's remove its public endpoints.

1. Log out of the VM.
2. In the Azure Management Portal, click VIRTUAL MACHINES on the left side and then click the VM that is in the point-to-site network.



3. Click ENDPOINTS at the top of the screen. Click the PowerShell endpoint and click DELETE at the bottom of the screen.
4. After it's finished, click the Remote Desktop endpoint and click DELETE at the bottom of the screen.
5. Run RDP on the local machine, and you will see that you can still connect to the VM. You are using the point-to-site virtual network that you set up.

Chapter 6

Databases

A persistent data store is at the heart of many applications. As you migrate existing applications to the Azure cloud, or create new applications, you will likely find yourself needing to interact with a database. The Azure platform provides several options from which to choose. You can choose from relational database offerings such as Azure SQL Database, SQL Server running in Azure Virtual Machines, or non-Microsoft databases such as Oracle or MySQL. If a nonrelational, NoSQL database fits your application needs better, services such as DocumentDB and Azure Table Storage might be a good fit. Furthermore, with Azure Virtual Machines you can install a wide range of database platforms (see Chapter 3, “Azure Virtual Machines,” for more information on Azure Virtual Machines).

When it comes time to determine a data storage approach for your application, the Azure platform offers a variety of database choices, allowing you to balance reduced friction and management versus fully customizable VMs.

Azure SQL Database

Azure SQL Database (formerly known as SQL Azure) provides a relational database-as-a-service, targeted at Online Transaction Processing (OLTP; i.e., data entry and retrieval transactions) workloads. This falls firmly in the PaaS category of cloud computing. Opting to use Azure SQL Database allows you to give up the physical management responsibilities of a database server but retain the vast majority of logical management and administrative responsibilities. Azure SQL Database provides many attractive features, such as elastic scale, predictable performance, business continuity, near-zero maintenance, and the use of familiar development languages and tools.

It is important to understand that with Azure SQL Database you do not get a physical server that you can manage. Because Azure SQL Database is a database-as-a-service, the underlying physical implementation details are outside your control.

Azure SQL Database is available in three tiers: Basic, Standard, and Premium. Within these tiers, performance is expressed in database throughput units (DTUs). A DTU is a synthetic measure that allows a quick comparison of the relative performance of the various database tiers. Within each tier, there are also performance levels (e.g., for Standard, S0, S1, and S2). These performance levels provide a way to increase or decrease the DTUs available within the tier. The maximum database size will vary across tiers, ranging from 2 GB to 500 GB. Table 6-1 lists some of the pertinent details for each of the database tiers.

Note There are actually two other tiers: Web and Business. The Web and Business tiers are currently deprecated, however, and are scheduled to be retired in September 2015. For the remainder of this

chapter, only the Basic, Standard, and Premium tiers are discussed unless otherwise noted.

TABLE 6-1 Azure SQL Database tiers and performance levels

Service Tier	Target Use Scenario	DTUs	Maximum Database Size	Performance Predictability/Variability
Basic	Small databases with minimal operations at a given time. Typically development or test usage scenarios.	5	2 GB	Good: Hour over hour
Standard	Cloud applications with multiple concurrent transactions.	S0 - 10 S1 - 20 S2 - 50 S3 - 100	250 GB	Better: Minute over minute
Premium	Mission-critical, enterprise-grade applications with high transaction rates and advanced business continuity features.	P1 - 100 P2 - 200 P3 - 800	500 GB	Best: Second over second

Determining which service tier to use will often depend on monitoring your application performance and then adjusting Azure SQL Database tiers. You can start with a Basic tier and then scale up to a Standard or Premium tier if needed. Adjusting service tiers and performance levels is an online operation, so you can continue to use the database while the operation completes. As you are adjusting tiers and performance levels, you might experience a temporary drop of database connections. Be sure to include retry logic in your application to be resilient to such transient errors.

See Also For more information on Azure SQL Database tiers and performance levels, including DTUs, please visit <http://msdn.microsoft.com/en-us/library/azure/dn741336.aspx>.

It is important to understand the relationship between an Azure SQL Database server and a database. When you create an Azure SQL Database server, you are creating a logical server. That logical Azure SQL Database server is essentially a Tabular Data Stream (TDS; the communication protocol between the client and SQL Server, or in this case, Azure SQL Database) endpoint (e.g., contoso.database.windows.net). That logical server will contain multiple Azure SQL database instances.

Creating a new Azure SQL Database server is a very quick operation. To create a new Azure SQL Database server using the Azure Preview Portal, click the green NEW button at the lower left of the portal to open the featured list of new Azure services you can create.

Select the SQL Database option to open the blade to create a new SQL database, as shown in Figure 6-1.

SQL database	
NAME	<input type="text" value="Enter database name"/>
PRICING TIER	Standard S0
COLLATION	SQL_Latin1_General_CI_AS
SERVER	Configure required settings
RESOURCE GROUP	Group
SUBSCRIPTION	Azure MVP MSDN Subscription

FIGURE 6-1 Create a new Azure SQL database.

On the SQL Database blade, you can enter several key pieces of information:

- **Name** Provide the name for the new database.
- **Pricing Tier** Select one of the available service tiers (Basic, Standard, or Premium) and associated performance levels.
- **Collation** Set the collation used for rules related to sorting and comparing data.
- **Server** Select an existing Azure SQL Database server or create a new server. When creating a new server, you will be able to provide the server name (e.g., contoso.database.windows.net), the administrative login and password, and the Azure region.
- **Resource Group** Select an existing or create a new logical group where the Azure SQL database will reside. Resource groups are helpful for grouping related Azure resources together.
- **Subscription** Select the desired Azure subscription.

When finished, click Create. It might take a few minutes for Azure to provision the new Azure SQL database. If you are creating a new database on an existing server, the new database will likely be ready within a few seconds.

As indicated in Table 6-1 earlier in this chapter, the maximum size for an Azure SQL Database instance is 500 GB. If your data needs exceed the capacity of a single database, you will need to use an alternative strategy to persist the necessary data. One such strategy is to spread the data across multiple databases, a process commonly referred to as database sharding. The ability to quickly create new database shards allows for elastic scale. Application owners can decide how and when to create new database shards to quickly scale out, thereby allowing an application to scale out across multiple databases. For more information on elastically growing and shrinking databases, please view the guidance on Azure SQL Database Elastic Scale at <http://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-scale-get-started/>. As of this writing, Azure SQL Database Elastic Scale is offered in a preview capacity.

Administration

One of the attractive features of Azure SQL Database is the near-zero maintenance it provides. Microsoft handles all patching, server configuration, load balancing, and database platform upgrades automatically. Additionally, Azure SQL Database handles management of system tables and filegroups automatically. You are required to perform common administrative tasks such as managing logical aspects of the database, including logins, tuning indexes, and query optimization.

When using Azure SQL Database, you can use the same tools, programming languages, and frameworks you are used to using with a traditional SQL Server. Azure SQL Database and SQL Server are similar in many ways, although not every SQL Server feature is available in Azure SQL Database (more details later in this chapter). However, the two do share one very important feature: both use TDS as the client protocol. This allows tools such as SQL Server Management Studio to connect to Azure SQL Database.

Firewall settings

Before you can connect to Azure SQL Database from any tool, including SQL Server Management Studio, you will likely need to adjust a firewall setting. Azure SQL Database comes preconfigured with firewall settings that will explicitly deny access from any IP address, even those originating from within Azure.

When creating a new Azure SQL Database server in the Azure Preview Portal, the default is to allow any Azure service (i.e., your Azure Website) in your subscription to access the server, as can be seen in Figure 6-2.

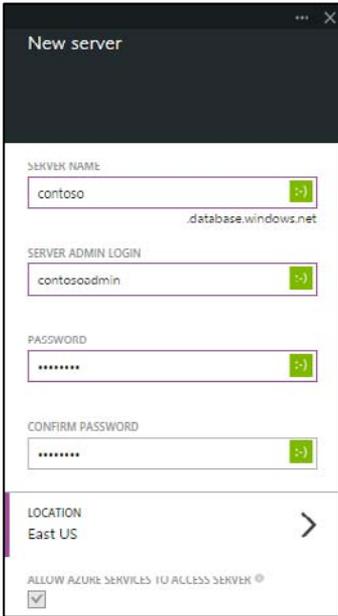


FIGURE 6-2 Create a new server.

To access the Azure SQL Database server from outside Azure—for example, from SQL Server Management Studio (SSMS)—you will need to modify the server firewall to allow access from the desired IP address (or range). On the Database blade, select the name of the SQL server in the Summary part, as seen in Figure 6-3.



FIGURE 6-3 Azure SQL Database summary part.

Doing so opens a new blade that allows you to manage various aspects of the Azure SQL Database server, including the firewall settings. Select the Firewall Settings part, as shown in Figure 6-4, to open the Firewall Settings blade.

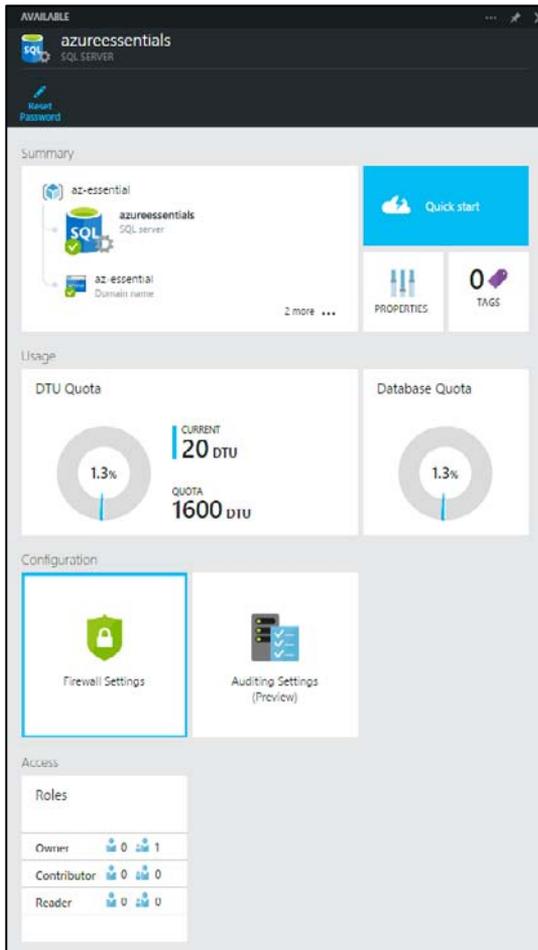


FIGURE 6-4 Azure SQL Database server settings.

On the Firewall Settings blade, you can enable or disable access to the server for Azure services or provide rules to allow access to the server from a specific IP address. If you are hosting other services in Azure (e.g., Azure Websites, Cloud Services, etc.) that need access to the Azure SQL Database instance, you will need to set ALLOW ACCESS TO AZURE SERVICES to ON. You can see an example of this in Figure 6-5.



FIGURE 6-5 The Firewall Settings blade.

Note It is also possible to set database-level firewall rules in addition to the server-level firewall rules available in the Azure Management Portal. Database-level firewall rules can be set programmatically via T-SQL statements. For more details, see <http://msdn.microsoft.com/en-us/library/azure/ee621782.aspx#ManagingRules>.

Connect using SQL Server Management Studio

Once you have added your IP address to the list of allowed IP addresses, open SQL Server Management Studio. You will need to know the full name of the Azure SQL Database server. You can find the full server name on the Properties blade, along with other important information, as shown in Figure 6-6. You can open the Properties blade by clicking the PROPERTIES part on the Database blade.

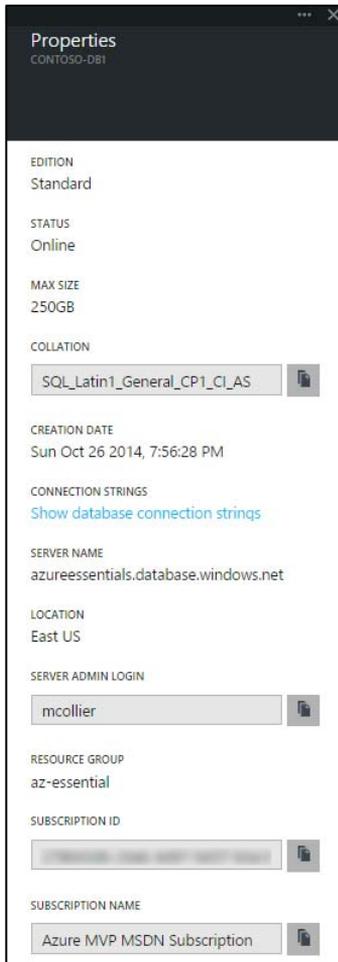


FIGURE 6-6 Database Properties blade.

When the Connect To Server dialog box in SQL Server Management Studio opens, as seen in Figure 6-7, enter the full server name, select SQL Server Authentication, and provide the administrative login and password you set when creating the database.

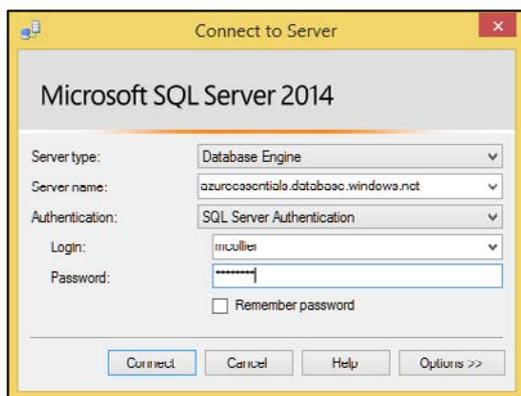


FIGURE 6-7 Connect to Azure SQL Database from SSMS.

Billing

Because Azure SQL Database is sold as a service, there is not a separate SQL Server license like you might find with on-premises SQL Server or with SQL Server in an Azure VM. Instead, you are charged per hour based on the highest Azure SQL Database service tier used during the hour (recall you can change tiers at any time). For example, if you start at 3 p.m. with an S1 and at 3:20 p.m. change to an S3, you are charged the S3 rate for the entire hour. The pricing change becomes effective when the change in tier or performance level is completed. For a detailed breakdown on Azure SQL Database pricing, please visit <http://azure.microsoft.com/en-us/pricing/details/sql-database/>.

In addition to understanding how Azure SQL Database is priced, it is important to understand how many databases you can get. After all, the number of databases you have will directly affect the price you pay. By default, there is a limit of six logical Azure SQL Database servers per subscription. Each server can host a maximum of 150 databases. The default limits are soft limits, and they can often be raised by submitting a support ticket with Azure Support.

Business continuity

Azure SQL Database provides several different options to address business continuity requirements. One way in which Azure SQL Database provides protection is through infrastructure redundancy. At any time in an Azure datacenter, there could be a hardware failure (hard drive, network, entire servers, etc.). Azure SQL Database provides high availability in the case of such hardware failures by keeping copies of the data on physically separate nodes. There are always three database nodes, or replicas, running: one primary and two secondary replicas. For write operations, data is written to the primary and one of the secondary replicas before the write transaction is considered complete. In the event of a failure, Azure SQL Database detects the failure and fails over to the secondary replica. If needed, a new

replica is then created.

Furthermore, business continuity with respect to databases often includes two categories: database recovery and disaster recovery. Database recovery refers to the ability to mitigate risk and recover from database corruption or an unintentional modification or deletion of data. To assist with database recovery, Azure SQL Database provides a feature called Point in Time Restore. Point in Time Restore allows you to restore a database to any previous point. The timeframe from which you can restore varies based on the selected Azure SQL Database tier: 7 days for Basic, 15 days for Standard, and 35 days for Premium.

To restore a database to a previous point, first select the desired database in the Azure Preview Portal and then click Restore, as shown in Figure 6-8.

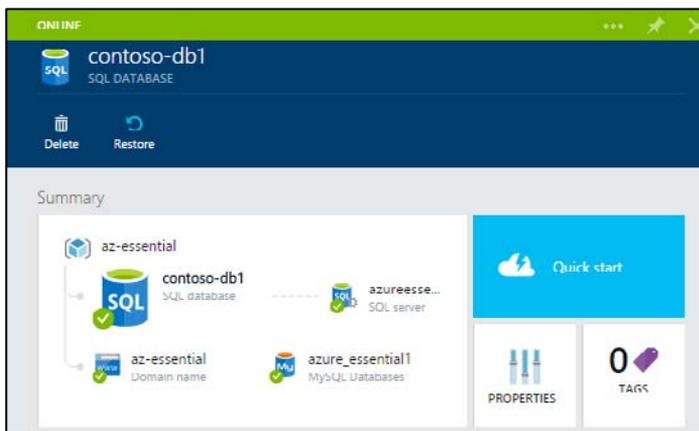


FIGURE 6-8 Option to restore a database.

Clicking the Restore button will open a new blade, shown in Figure 6-9, that will allow you to enter the name for the restored database (or keep the autogenerated default name) and the restore point (date and time, at one-minute intervals).

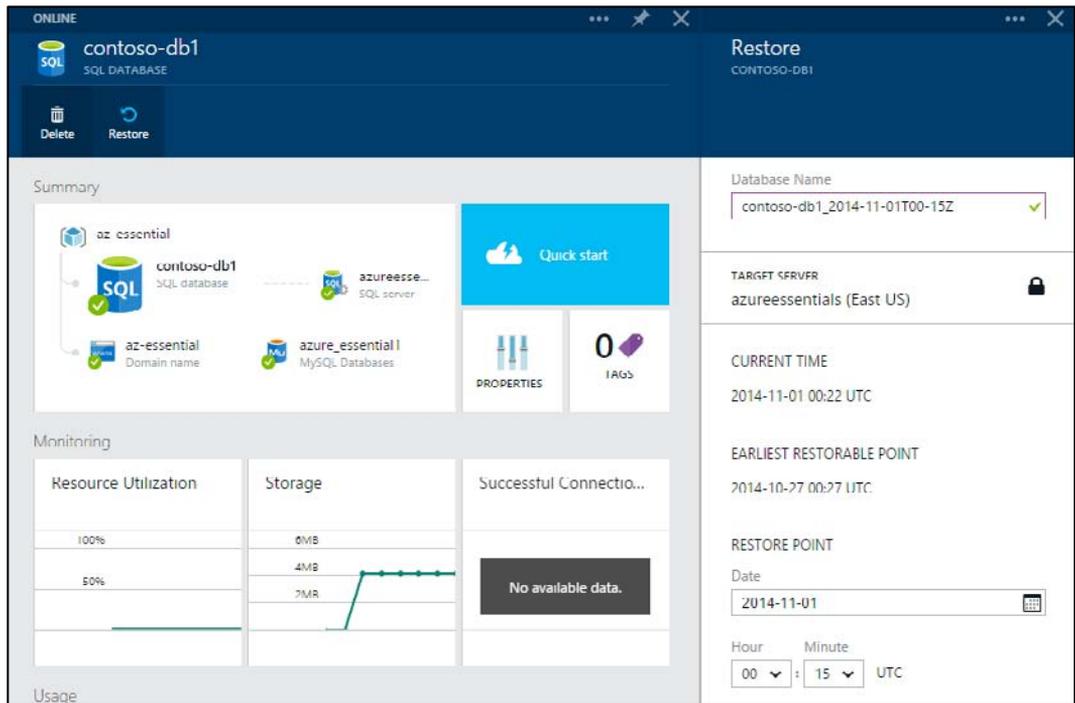


FIGURE 6-9 Database restore settings.

The restore operation could take a long time to complete. The exact time to restore can be difficult to predict, as it depends on several factors, including the size of the database, the restore point time selected, and the activity log that needs to be replayed to get to the restore point. For some large databases, this process could take several hours. You can monitor the completion status from the main Notifications blade in the portal.

If you have deleted a database, you can restore the entire database. To do so, first select the Azure SQL Database server that contained the database and then select the Operations part from the SQL Server blade. This will open a new Deleted Databases blade, as shown in Figure 6-10.

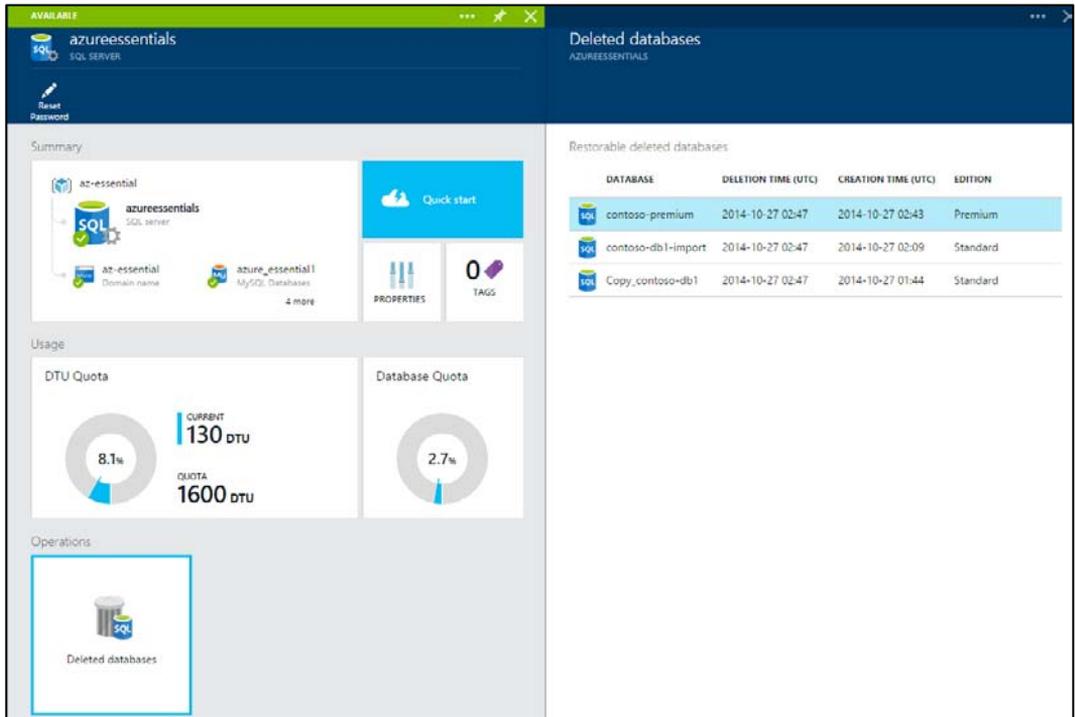


FIGURE 6-10 Restoring a deleted database.

If there are multiple databases that were deleted, select the database to restore. On the resulting Restore blade, provide a name for the database to be restored, as shown in Figure 6-11. The database can only be restored to the point at which it was deleted. After you click Create, the restore request will be submitted. Just like a point-in-time restore, the process to restore a deleted database could take a long time to complete.

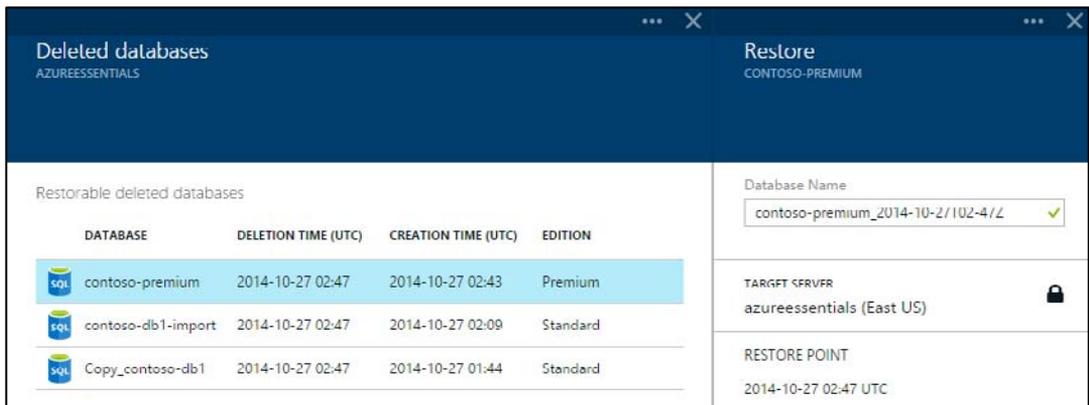
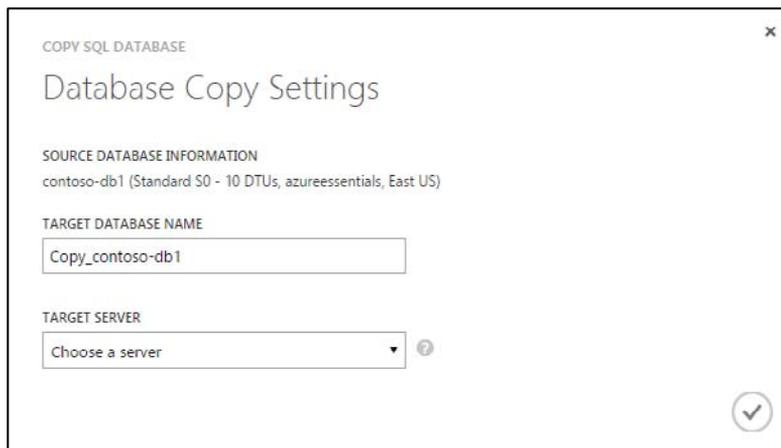


FIGURE 6-11 Settings for restoring a deleted database.

Point in Time Restore is helpful when there is a need to recover a database to a known good point, which is often necessary because of user error. However, this is only one aspect of business continuity, the other being disaster recovery. Disaster recovery refers to the ability to restore operations to a working state in the event a disaster renders the primary region unrecoverable. Azure SQL Database provides several features that can be helpful in preparing a disaster recovery plan: Database Copy and Export, Standard and Active Geo-Replication, and Geo-Restore.

Database Copy

Database Copy creates a copy of the desired database on the same Azure SQL Database server or on another server in another Azure region. The copy is transactionally consistent with the source when the copy operation completes. To perform a database copy, you will need to use the Azure Management Portal. Select the desired database and then click the COPY button on the bottom command bar to open the Database Copy Settings dialog box, which will allow you to provide the Target Database Name and the Target Server, as seen in Figure 6-12. You can monitor the Database Copy progress from the DATABASES list in the portal.



COPY SQL DATABASE

Database Copy Settings

SOURCE DATABASE INFORMATION
contoso-db1 (Standard S0 - 10 DTUs, azureessentials, East US)

TARGET DATABASE NAME

TARGET SERVER

FIGURE 6-12 Database Copy Settings dialog box.

Import and Export

Import and Export allows you to export an Azure SQL database as a BACPAC file. The BACPAC file is saved to Azure Blob storage. You can then import the BACPAC file to a new server, thus creating a copy of the source from the point when the BACPAC was created. It should be noted that the export process does not guarantee a transactionally consistent copy of the source database. Therefore, it is recommended that you first create a copy (which does provide transactional consistency) and then perform the export.

Note Keep in mind that a BACPAC file is saved as a block blob in Azure Blob storage. Block blobs have a maximum size of 200 GB. As such, if you have database larger than 200 GB, you might not be able to export the database as a BACPAC file. The exact maximum size varies because the BACPAC file is a compressed copy (schema and data) of the database.

To export an Azure SQL database, you will need to use the Azure Management Portal. After navigating to the SQL Databases section, select the database you would like to export. In the DASHBOARD section, click EXPORT on the bottom command bar, as seen in Figure 6-13.

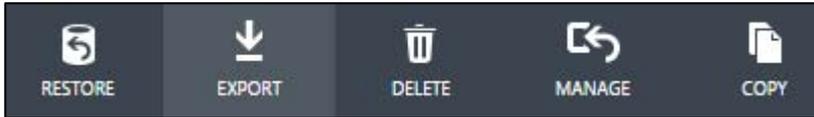


FIGURE 6-13 Export an Azure SQL database.

In the resulting Export Database Settings dialog box, provide a name for the exported BACPAC file, select an existing Azure Storage account or create a new account, and provide the administrative login and password for the Azure SQL Database server, as shown in Figure 6-14.

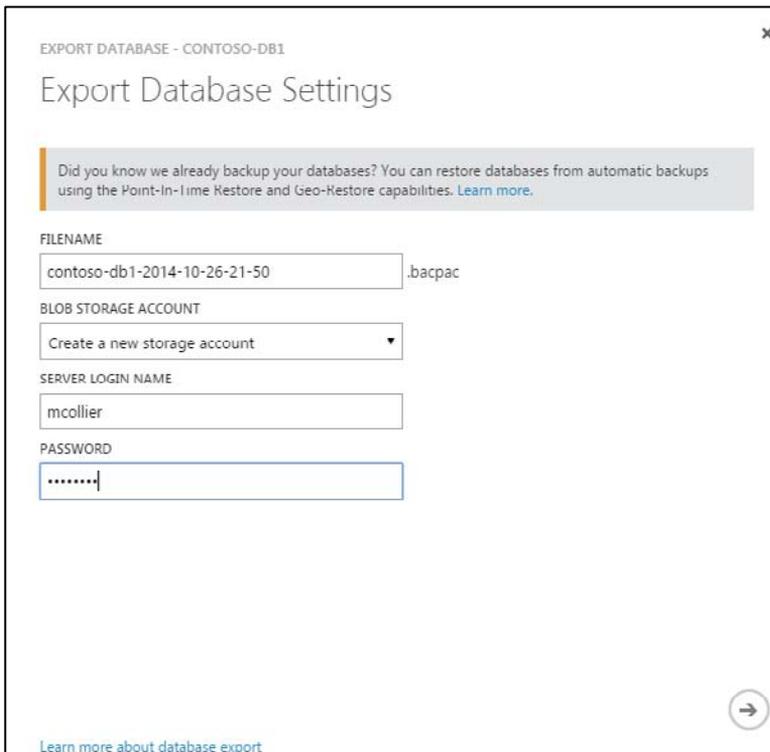


FIGURE 6-14 Export an Azure SQL database.

In addition to performing on-demand exports, you can configure and schedule automatic exports. In the Azure Management Portal, select the desired database and go to the CONFIGURE section. Change the EXPORT STATUS from NONE to AUTOMATIC. As shown in Figure 6-15, provide the name of the Azure Storage account to use, the export frequency, the retention period, and the database credentials.

The screenshot shows the 'Automated Export' configuration page for a database named 'contoso-db1'. The page has a navigation bar with links for DASHBOARD, MONITOR, SCALE, CONFIGURE, GEO-REPLICATION, and AUDITING & SECURITY (PREVIEW). A notification banner at the top asks if the user knows they already backup their databases and offers to restore from automatic backups. The main configuration area is titled 'automated export' and includes the following settings:

- EXPORT STATUS:** Buttons for NONE, AUTOMATIC (highlighted in purple), and PREVIEW. A help icon (?) is next to PREVIEW.
- STORAGE ACCOUNT:** A dropdown menu showing 'azureessential'.
- FREQUENCY:** 'Every' 7 Days. A help icon (?) is next to 'Days'. Below this, 'Start date' is set to '2014-10-26' and 'Time' is '12:00 AM'. A note indicates 'Local Time (UTC-04:00)' with a help icon (?) next to it.
- RETENTION:** '30' Days. A help icon (?) is next to 'Days'. Below this, a checkbox 'Always keep at least one export file.' is checked, with a help icon (?) next to it.
- SERVER LOGIN NAME:** A text input field containing 'mcollier'.
- SERVER LOGIN PASSWORD:** A password input field with a green checkmark icon to its right, indicating it is valid.

At the bottom of the page, there are two buttons: 'SAVE' and 'DISCARD'.

FIGURE 6-15 Automatic Database Export settings.

You can use the exported database to create a new Azure SQL database. From the Azure Management Portal, click NEW in the lower-left corner and then proceed to DATA SERVICES, then SQL DATABASE, and finally IMPORT, as seen in Figure 6-16.



FIGURE 6-16 Azure SQL Database import.

In the resulting Specify Database Settings dialog box, you will need to browse to the BACPAC file in Azure Blob storage and provide a name for the new database, along with selecting the target Azure subscription, server, and performance level, as seen in Figure 6-17.

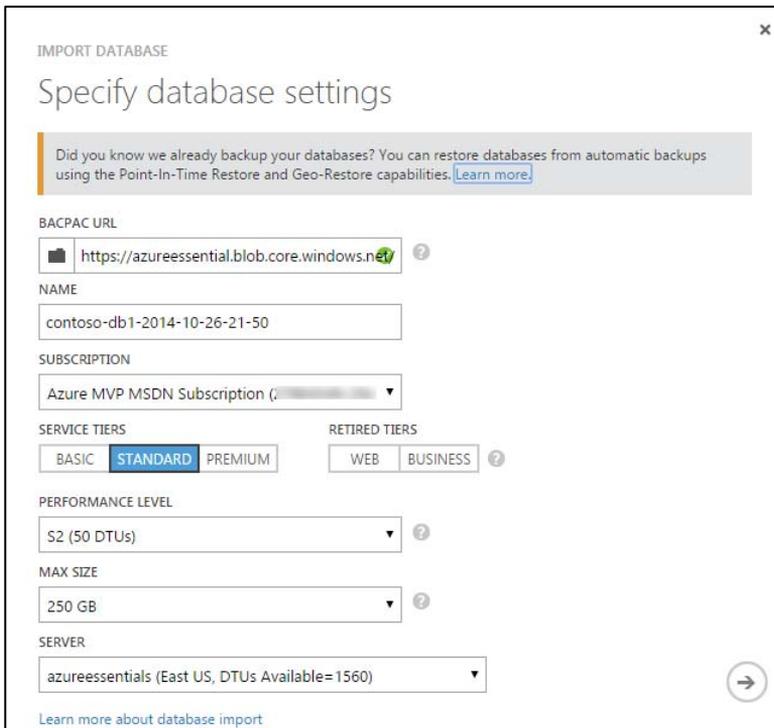


FIGURE 6-17 Azure SQL Database Import settings.

Note For Basic, Standard, and Premium databases, Point in Time Restore is the recommended approach for recovering from database corruption or accidental data loss. It should be noted that Point in Time Restore is not available for Web or Business edition databases. If you need to keep a backup copy of the database available for longer than the Point in Time Restore retention period (7 days for Basic, 15 days for Standard, and 35 days for Premium), then using Database Copy and Export is still going to be your best approach.

Standard Geo-Replication

Standard Geo-Replication allows you to create a single offline secondary database in the paired datacenter of the primary database. The secondary database is unavailable for client connections until the region hosting the primary database fails. The secondary database is charged at 75 percent of the primary (e.g., an S1 costs approximately \$30.00 per month and the secondary would cost approximately \$22.50 for the month, for a total of \$55.50 for the month).

Active Geo-Replication

Active Geo-Replication, which is a feature available only for Premium tier databases, allows you to create up to four readable secondary databases across multiple Azure regions. It is up to you to determine when to fail over one of the secondary databases (unlike Standard Geo-Replication). Each readable secondary is charged at the same rate as the primary.

To enable Standard or Active Geo-Replication, use the Azure Preview Portal and select the desired database. From the Geo Replication part, shown in Figure 6-18, you will be able to see a map displaying any existing secondary database or an option to configure geo-replication if none has been configured.

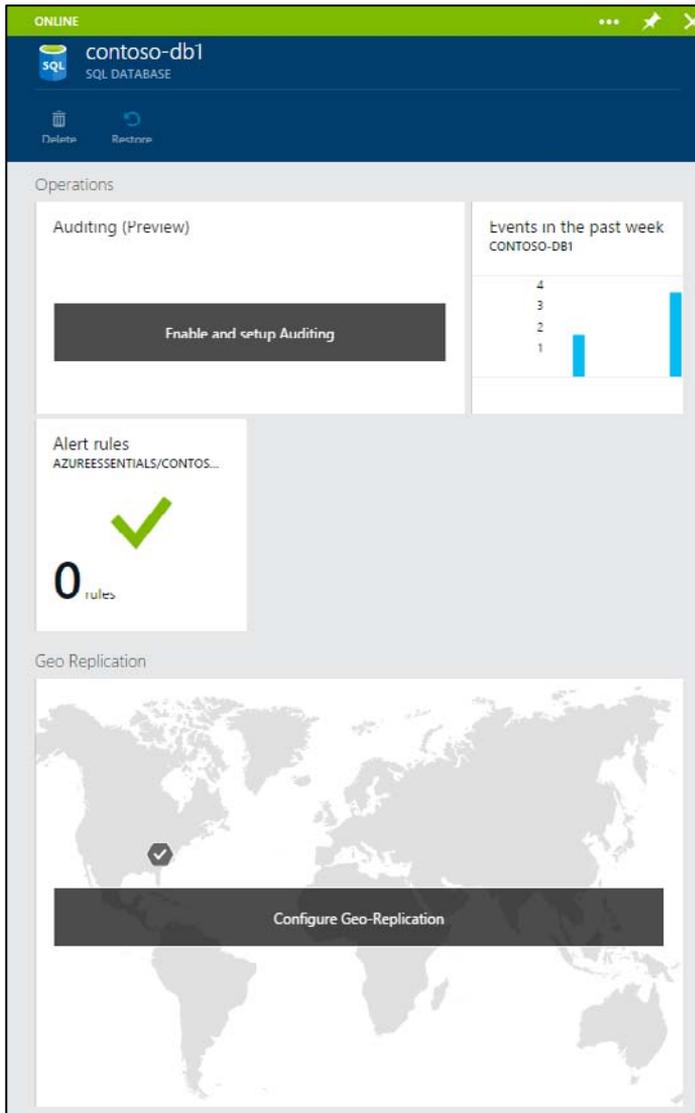


FIGURE 6-18 Configure Geo-Replication.

Select the Geo Replication part (displaying the map) to open a new Geo Replication blade. On this blade, you can view all of the potential secondary locations and then select the desired location. Note that for a Standard database, only the paired region will be available. However, from a Premium

database, you can select from any of the available locations.

Click the location in the map, as seen in Figure 6-19, or select from the list of TARGET REGIONS as displayed on the blade. This will open a new Create Secondary blade to allow you to configure properties related to the secondary database.

The screenshot shows the 'Geo Replication' blade with the 'Create Secondary' configuration panel. The blade title is 'Geo Replication' and the panel title is 'Create Secondary'. A sub-header reads: 'Create geo-replicated secondaries to protect against prolonged datacenter outages. Secondaries have price implications. [Learn more](#)'. The configuration panel includes the following options:

- REGION: West US (locked)
- Database Name: contoso-db1
- PRICING TIER: S0 Standard (10 DTUs) (locked)
- SECONDARY TYPE: Non-readable
- SERVER NAME: Configure required settings

The main area of the blade features a world map with several regions highlighted. Below the map is a table with columns 'SERVER/DATABASE' and 'STATUS'.

SERVER/DATABASE	STATUS
PRIMARY	
East US	Online
SECONDARIES	
<i>Geo-replication is not configured; select a region below to add a secondary.</i>	
TARGET REGIONS	
Brazil South	
Central US	
East Asia	
East US	

A 'Create' button is located at the bottom right of the configuration panel.

FIGURE 6-19 Geo-replicated secondary configuration settings.

For Standard Geo-Replication, the SECONDARY TYPE will default to Non-Readable only. For Active Geo-Replication, the SECONDARY TYPE will default to Readable and you will be able to select from multiple Azure regions for the REGION.

Geo-Restore

Finally, the Geo-Restore feature in Azure SQL Database allows you to restore an Azure SQL database from a backup to any Azure SQL Database server in any Azure region. The time to restore will vary based on size of the database, performance level, and number of concurrent restore requests in the target Azure region. Both Point in Time Restore (as discussed earlier in this chapter) and Geo-Restore are possible because Azure SQL Database automatically creates backups of every database. Full backups are performed once per week, differential backups once a day, and transaction log backups every five minutes. The backup data is persisted in Azure Blob storage (RA-GRS) in a geo-redundant paired region (e.g., East US and West US, North Europe and West Europe, etc.), as seen in Figure 6-20.

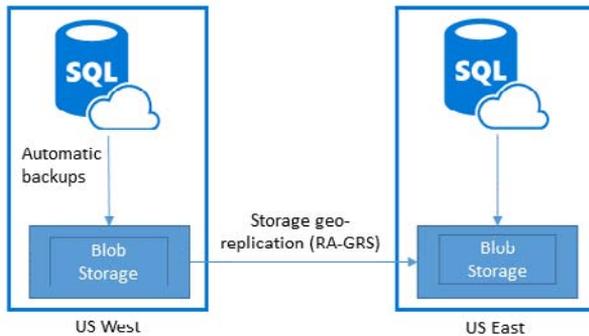


FIGURE 6-20 Azure SQL Database configured to automatically back up to geo-replicated storage.

At the time of this writing, performing a geo-restore is available only in the Azure Management Portal (manage.windowsazure.com).

First, navigate to the desired Azure SQL Database server and select the server that you wish to restore. Then, on the server page, select BACKUPS from the top menu, as seen in Figure 6-21, and click RESTORE on the bottom command bar.

The screenshot displays the 'azureessentials' interface. On the left is a navigation sidebar with a back arrow icon and a list of server names: 'eczshzk824', 'rtarg4lhmd', and 'y8diqikuve'. The top navigation bar includes 'DASHBOARD', 'DATABASES', 'CONFIGURE', 'HISTORY', and 'BACKUPS'. The main content area is titled 'GEO-REDUNDANT BACKUPS' and contains a table with the following data:

NAME	LAST BACKUP TIME (UTC)	EDITION
contoso-db1_2014-10-27T01-18Z	2014-10-31 17:21	Premium
Copy_contoso-db1	2014-10-27 01:51	Standard
contoso-db1-import	2014-10-27 02:14	Standard
contoso-premium	2014-10-27 02:45	Premium
contoso-db1_2014-11-01T00-15Z	2014-11-01 00:36	Standard
azureessential-db1	2014-10-31 18:10	Standard
contoso-db1	2014-10-31 19:27	Standard

At the bottom of the page, there is a 'NEW' button on the left and a 'RESTORE' button with a circular arrow icon on the right.

FIGURE 6-21 Geo-redundant backups.

In the resulting Specify Restore Settings dialog box, shown in Figure 6-22, provide the new database name and select the target server (either select an existing one or create a new one). Click the check mark to submit the restore request.

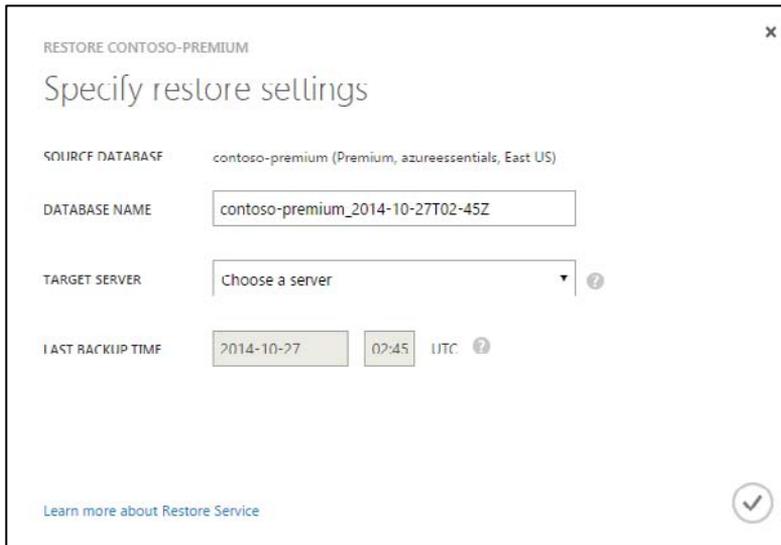


FIGURE 6-22 Specify Restore Settings for geo-restore.

The recovery time objective (RTO) and recovery point objective (RPO) are key factors in determining which feature(s) to use for your disaster recovery plan. The RTO indicates the maximum downtime before the application is functional after a disaster. The RPO indicates the maximum amount of recent data loss (in terms of time) before the application is functional after a disaster. The business continuity features for Azure SQL Database across the service tiers, including RTO and RPO, are depicted in Table 6-2.

TABLE 6-2 Business continuity options for Azure SQL Database tiers

Business Continuity Feature	Basic Tier	Standard Tier	Premium Tier
Point in Time Restore	Last 7 days	Last 14 days	Last 35 days
Geo-Restore	RTO < 24 hours RPO < 24 hours	RTO < 24 hours RPO < 24 hours	RTO < 24 hours RPO < 24 hours
Standard Geo-Replication	N/A	RTO < 2 hours RPO < 30 minutes	RTO < 2 hours RPO < 30 minutes
Active Geo-Replication	N/A	N/A	RTO < 1 hour RPO < 5 minutes

Service Level Agreement

Microsoft provides a 99.99 percent database connectivity SLA for Basic, Standard, and Premium tiers. The SLA applies to being able to connect to the database only, not to any performance targets with respect to the various tiers.

Applications connecting to Azure SQL Database

When writing applications that need to connect to Azure SQL Database, you can use popular programming languages such as .NET, PHP, Java, and many more. Entity Framework, starting with .NET Framework 3.5 Service Pack 1, is also supported. One of the first things you will need is the connection string. You can obtain the connection string from the Azure Preview Portal by clicking the PROPERTIES part on the desired database blade, as seen previously in Figure 6-6. Clicking the Show Database Connection Strings link under CONNECTION STRINGS will open a new Database Connection Strings blade, as seen in Figure 6-23, displaying the connection string in multiple formats, including ADO.NET, PHP, JDBC, and ODBC (which works for node.js applications).

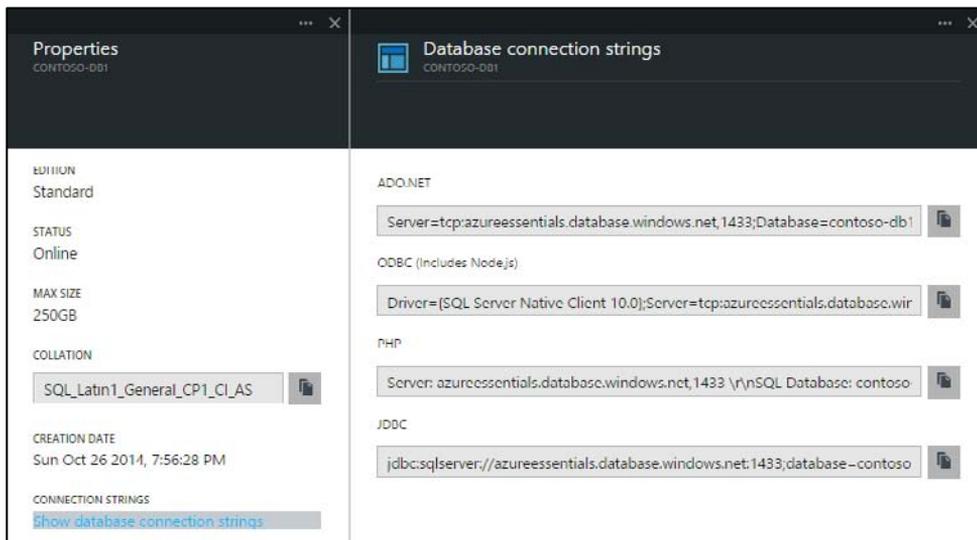


FIGURE 6-23 Database Connection Strings blade.

The connection string for Azure SQL Database is similar to what you would use for SQL Server. For example, for ADO.NET, the connection string format is as follows:

```
"Server=tcp:{your_servername_here}.database.windows.net,1433;Database={your_database_name_here};
User ID={your_username_here}@{your_servername_here};
Password={your_password_here};Trusted_Connection=False;Encrypt=True;Connection Timeout=30;"
```

Note that the connection string sets the `Trusted_Connection` property to `False` and the `Encrypt` property to `True`. This is to provide additional protection while accessing Azure SQL Database over the Internet. Doing so helps thwart potential man-in-the-middle attacks. Azure SQL Database will actually force the connection to be encrypted regardless of the setting.

When writing code against Azure SQL Database, it is important to defend your code against transient errors. Transient errors are errors that are intermittent and will likely be resolved if the command is retried. These errors are more common with Azure SQL Database than with databases

accessed via a local area network (LAN). This is due to the inherently unreliable network that is the Internet and the fact that as a managed service, Azure SQL Database might periodically undergo maintenance activities that could cause connections to temporarily drop. Applications should plan for, and defend against, transient errors by incorporating retry logic when creating connections or executing commands against Azure SQL Database. For .NET applications using Entity Framework, Entity Framework 6 contains connection resiliency/retry logic that will detect transient errors from Azure SQL Database and retry the command. For other .NET applications, Enterprise Library 5 and 6 (see <http://msdn.microsoft.com/en-us/library/ff648951.aspx>) from Microsoft Patterns & Practices contain an application block called the Transient Fault Handling Application Block. This library can also be used to detect transient errors and retry commands.

SQL Server in Azure Virtual Machines

Whereas Azure SQL Database provides database-as-a-service with virtually no administration and with enterprise-grade features, there are still situations in which running your own SQL Server deployment is necessary. There are numerous reasons to host your own SQL Server deployment. A common reason is the requirement to use features that are not available in Azure SQL Database.

As discussed in Chapter 3, Azure Virtual Machines provides the ability to host and manage your own VMs. What you use the VM for is largely your responsibility, and this includes using it to install, configure, and manage your own full SQL Server VM or cluster of VMs.

Billing

When running your own SQL Server deployment on Azure Virtual Machines, there are three important cost factors to understand. First, there is the cost of the Windows VM itself. Recall that Azure VMs are charged on a per-minute usage model. Second, there is the SQL Server license cost. When using a SQL Server image from the Azure Virtual Machines image gallery, you will pay an additional per-minute SQL Server license cost, which varies according to the version of SQL Server (Web, Standard, or Enterprise) and the target size of the VM. Finally, you will also pay for the Azure Storage cost. Azure Storage (specifically page blobs) is used as the persistence mechanism for Azure Virtual Machines disks. To summarize, the cost for SQL Server in Azure Virtual Machines can be represented as Total Cost = Windows Server cost + SQL Server license cost + Azure Storage cost.

If you have your own SQL Server license, you can use that instead of paying the per-minute charge associated with using a SQL Server license obtained from an Azure Virtual Machines image. In this case, you pay only for the Windows Server license and any related Azure Storage costs. The ability to use your own SQL Server license is a feature of License Mobility through Microsoft's Software Assurance on Azure program. For more information, see <http://azure.microsoft.com/pricing/license-mobility/>.

Virtual machine configuration

When configuring SQL Server in Azure Virtual Machines, there are a few important considerations to take into account.

- VM considerations
 - Use an A2 VM or higher for SQL Server Standard Edition.
 - Keep the storage account and VM in the same Azure region to minimize latency. Additionally, if using multiple data disks to store database data and log files, disable geo-replication because consistent write order across disks is not guaranteed.
- Disk considerations
 - Use a data disk with the cache policy set to None for placement of database data and log files.
 - Do not store database data and log files on the D drive. The D drive is a physical temporary disk and is not persisted to Azure Blob storage. However, if you are using a D-series VM, you might consider storing the tempdb database on the D drive. D-series VMs use an SSD drive for the D drive, and thus tempdb performance could be improved.
 - For workloads that exceed the 500 IOPS limit for a single data disk, you may attach multiple data disks (up to the maximum allowed by the VM size) and use disk striping as a way to increase IOPS.

See Also For a comprehensive review of performance best practices for SQL Server in Azure Virtual Machines, please read the MSDN guidance at <http://msdn.microsoft.com/en-us/library/azure/dn133149.aspx>.

Business continuity

Most of the high availability and disaster recovery (HADR) solutions you might run for on-premises SQL Server deployments are also available when running SQL Server in Azure Virtual Machines. But why do you need to be concerned about HADR for SQL Server in Azure Virtual Machines? As discussed in Chapter 3, Azure provides high-availability features for the VMs, but not necessarily for SQL Server running on the VM. It is entirely possible for the VM to be online but the SQL Server instance to be offline, unhealthy, or both. Additionally, it is possible for the VM to be unavailable due to hardware failure or software upgrades. As such, a practiced HADR strategy should be considered.

SQL Server in Azure Virtual Machines supports many of the same HADR technologies that are available for on-premises SQL Server deployments: AlwaysOn, database mirroring, log shipping, and backup and restore to Azure Blob storage (available in SQL Server 2012 and SQL Server 2014). Depending on the technology used, it might be possible to establish a hybrid topology to allow the HADR technology to span between an Azure region and an on-premises datacenter. Some options, such as SQL Server AlwaysOn, allow for a topology that can even span multiple Azure regions.

See Also For more detailed information on how to configure various HADR solutions with SQL Server in Azure Virtual Machines, please refer to the guidance on MSDN at <http://msdn.microsoft.com/en-us/library/azure/jj870962.aspx>.

Comparing Azure SQL Database with SQL Server in Azure Virtual Machines

The decision to use Azure SQL Database or SQL Server in Azure Virtual Machines can be difficult. On the one hand, Azure SQL Database is ideal for reducing the administrative cost related to provisioning and managing relational databases, because many tasks such as upgrades, patching, backups, and business continuity scenarios are handled automatically. On the other hand, SQL Server in Azure Virtual Machines provides the option to migrate or extend existing on-premises SQL Server workloads to Azure. Even though there are additional administrative costs with running SQL Server in Azure Virtual Machines, the ability to maintain fine-grained control over those tasks could be worthwhile for some users and scenarios.

Although Azure SQL Database and SQL Server are similar in many areas, there are some key differences, most notably in SQL Server features that are not currently supported in Azure SQL Database, such as the following:

- Windows authentication.
- Distributed transactions.
- FILESTREAM Data.
- Full-Text Search.
- User-Defined Types (UDTs).
- Transparent Data Encryption (TDE).
- Database mirroring.
- Extended stored procedures.
- SQL Server Agent/Jobs.
- SQL Server Reporting Services (SSRS) and SQL Server Integration Services (SSIS) are not supported. Alternatively, run an SQL Server on-premises or in an Azure VM and connect to an Azure SQL database.
- T-SQL features
 - USE statement is not supported. To change databases, a new connection must be established.

- Common Language Runtime (CLR).
- Distributed (multipart) queries.

There are also some general limitations and requirements imposed by Azure SQL Database that are different from a traditional SQL Server, such as the following:

- All Azure SQL Database tables must have a clustered index. Insert operations will be denied until a clustered index is created.
- Connections that are idle for more than 30 minutes are automatically terminated.
- Azure SQL Database accepts connections only on port 1433 (a common port for SQL Server). You should ensure your firewall allows outbound TCP connections on port 1433.
- Due to different implementations of the TDS protocol, some tools might need to append the server name to the login: `[login]@[servername]` (e.g., joe@x45r4dj).

Note The limitations just listed are only a few of the limitations that you should be aware of when working with Azure SQL Database. For a complete list, please see the guidance on MSDN at <http://msdn.microsoft.com/en-us/library/azure/ff394102.aspx>. You can also find the related T-SQL statement references at the following locations:

Supported <http://msdn.microsoft.com/en-us/library/azure/ee336270.aspx>

Partially Supported <http://msdn.microsoft.com/en-us/library/azure/ee336267.aspx>

Unsupported <http://msdn.microsoft.com/en-us/library/azure/ee336253.aspx>

There are many factors to consider when choosing between Azure SQL Database and SQL Server in Azure Virtual Machines: database size, existing application versus new application, level of administrative control (including hardware infrastructure), business continuity strategy, and hybrid scenarios, just to name a few. Azure SQL Database is often the right solution for cloud-designed applications that are not using unsupported features and for which near-zero administration is a key priority. SQL Server in Azure Virtual Machines is often the right choice for new or existing applications that require a high level of control and customization (i.e., full compatibility with SQL Server) and for which there is a desire to no longer maintain on-premises hardware.

Database alternatives

MySQL

Another popular relational database is MySQL. Microsoft has collaborated with SuccessBricks to bring SuccessBricks' ClearDb database-as-a-service for MySQL to the Azure platform.

To get started, open the Azure Preview Portal and click the green NEW button. Find the MySQL Database feature in the list of available services, as shown in Figure 6-24.



FIGURE 6-24 Creating a new MySQL database.

When the New MySQL Database blade opens (Figure 6-25), you will have the opportunity to provide important details about your new MySQL database, including the following:

- **Database Name** The name for the new database.
- **Pricing Tier** Select one of the available pricing tiers. These tiers are not related in any way to the Azure SQL Database tiers or performance levels.
- **Resource Group** Select an existing, or create a new, logical group where the Azure SQL Database will reside. Resource groups are helpful for grouping related Azure resources together.
- **Subscription** The desired Azure subscription.
- **Location** The desired Azure region.
- **Legal Terms** Agree to the legal terms, which detail that the service is provided by SuccessBricks and not Microsoft, to continue.

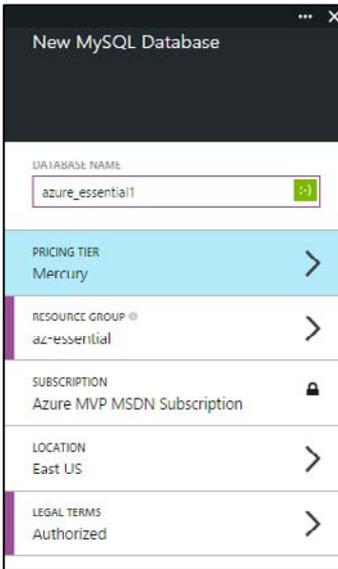


FIGURE 6-25 New MySQL Database blade settings.

When finished, click the blue Create button at the bottom of the blade to submit the request to create the new MySQL database. Once the database is created, the database blade will automatically open. Clicking the Properties part will open a new blade, as seen in Figure 6-26, allowing you to view details about the new MySQL Database, such as the full hostname, username, password, connection string, and more.

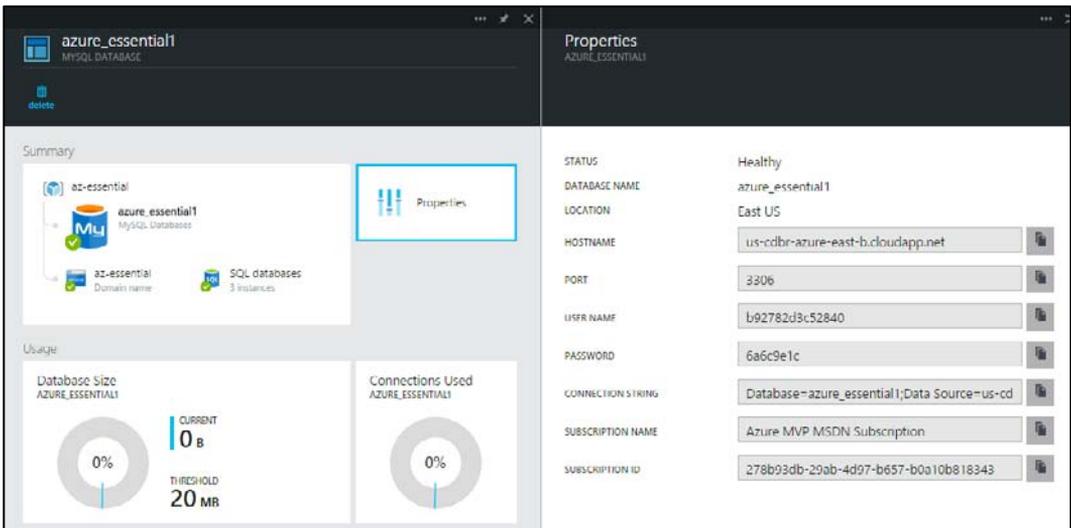


FIGURE 6-26 MySQL Database Properties blade.

NoSQL options

You do not have to use Azure SQL Database or SQL Server in Azure Virtual Machines for your database needs. Azure provides two NoSQL options: DocumentDB and Azure Table storage.

DocumentDB

DocumentDB is a fully managed, highly scalable, NoSQL document database service available on the Azure platform. DocumentDB was designed to natively support JSON documents, and it automatically indexes all JSON documents added to the database. You use familiar SQL syntax to query the documents. DocumentDB is currently in Preview as of the time of this writing.

See Also For more information on DocumentDB, please see <http://azure.microsoft.com/en-us/services/documentdb/>.

Table storage

Azure Table storage is a cost-effective, highly scalable, key/value NoSQL store available on the Azure platform. Table storage is capable of storing up to 500 TB per storage account, with a single subscription supporting 100 storage accounts. Table storage is a semistructured NoSQL data store that uses two keys, a Partition Key and Row Key, as the primary composite index for the table.

See Also For more information on Azure Table storage, please see <http://azure.microsoft.com/en-us/documentation/articles/storage-introduction/>.

Chapter 7

Azure Active Directory

Our identity defines who we are and what we do. Identity is at the heart of many applications and services. Identity tells the story of who used the application and what actions the user can perform. Without identity, applications often lose the closeness, or personal relationship, so many users find appealing. Without identity, application administrators have a more difficult time determining who used their application and what actions the users were able to perform.

The identity story in the Microsoft Azure platform centers on Azure Active Directory (Azure AD). Azure AD provides a cloud-friendly, secure, scalable, modern identity solution that can serve cloud-hosted and on-premises solutions alike.

Overview of Azure Active Directory

Before discussing what Azure AD is, it can be helpful to understand what it is not. First, Azure AD is not a replacement for Windows Server Active Directory. As it exists today, you are not able to domain join machines (physical or virtual) to Azure AD. You cannot allocate objects such as printers to Azure AD. If you need the full capabilities of Windows Server Active Directory, consider installing and configuring Windows Server Active Directory on Azure Virtual Machines.

What is Azure Active Directory?

Azure AD is a robust, secure, multitenant directory service that provides identity and access management in the cloud. In fact, Azure AD is the directory store for many of Microsoft's premium cloud services, such as Microsoft Office 365, Microsoft Dynamics CRM Online, Windows Intune, and, of course, Microsoft Azure. Much like Windows Server Active Directory provides identity and access management for on-premises solutions, Azure AD does so as a service available in Azure. However, instead of you assuming the responsibility of provisioning and configuring the multiple servers necessary for on-premises Active Directory, Microsoft is responsible for managing the entirety of the Azure AD infrastructure (high availability, scalability, disaster recovery, and so on). As a consumer of the Azure AD service (directory-as-a-service), you decide what users and which of their related information should reside in the directory, who can use the information, and what applications have access to the information.

Azure AD should not be considered a replacement for Window Server Active Directory. Instead, Azure AD is a complementary service. If you already have Active Directory on premises, the users and groups can be synchronized to your Azure AD directory by using a tool such as Azure Active Directory Sync (AADSync). The precursor to AADSync was the DirSync tool.

Azure AD can be associated with an on-premises Active Directory to support single sign-on (SSO). This can be either true SSO using Active Directory Federation Services (AD FS) to federate the on-premises identity to Azure AD or shared sign-on, in which AADSync is used to sync a password hash between Active Directory and Azure AD. Shared sign-on is simpler to configure at the cost of a small delay in the synchronization of password changes because, by default, AADSync replicates changes every three hours.

Azure AD is a multitenant directory service. Each tenant is a dedicated instance of Azure AD that you own when you sign up for a Microsoft cloud service (Azure, Office 365, and so on). Each tenant directory is isolated from the others in the service and designed to ensure user data is not accessible from other tenants, meaning others cannot access data in your directory unless an administrator grants explicit access.

It is important to note that Azure AD is not just for cloud or Azure-hosted solutions. Azure AD can be used by both cloud (hosted in Azure or elsewhere) and on-premises solutions. Instead of using technologies like Kerberos or Lightweight Directory Access Protocol (LDAP) to access Active Directory (as you would on premises), Azure AD is accessible via a modern REST API. This allows a wide range of applications—on-premises, cloud, mobile, and so on—to access the rich information available in the Azure AD directory. For developers, this opens up a vast opportunity that previously, with on-premises solutions, either wasn't possible or was difficult to achieve. By leveraging Azure AD and its Graph REST API, developers are able to easily establish SSO for cloud applications and to query and write (create, update, delete) against the directory data.

Active Directory editions

As of this writing, there are three tiers for Azure AD:

- **Free** Provides the ability to manage users, synchronize with on-premises Active Directory, establish SSO across Azure and Office 365, and access SaaS applications in the Azure AD application gallery.
- **Basic** Provides all the features of the Free tier, plus self-service password resets, group-based application access, customizable branding, and a 99.9 percent availability SLA.
- **Premium** Provides all the features of the Free and Basic tiers, plus self-service group management, advanced security reports and alerts, Multi-Factor Authentication, licenses for Microsoft Forefront Identity Manager (Microsoft Identity Manager), and future enterprise features such as password write-back.

See Also For more details on the Azure AD tiers, please refer to <http://msdn.microsoft.com/library/azure/dn532272.aspx>.

Creating a directory

It is easy to create your own Azure AD directory. In fact, as mentioned earlier, if you are using a Microsoft cloud service such as Office 365, you already have an Azure AD directory. You can associate a new Azure subscription with an existing directory used for authenticating with other Microsoft cloud services. To do so, sign into the Azure Management Portal (<http://manage.windowsazure.com>) using your existing work or school account (formerly known as an organizational account). If there are no existing Azure subscriptions associated with your account, the portal will return a message indicating there are no subscriptions associated with your account, as seen in Figure 7-1. You will need to sign up for an Azure subscription. Once signed up for an Azure subscription, you will be able to use your work or school account to access the Azure subscription.

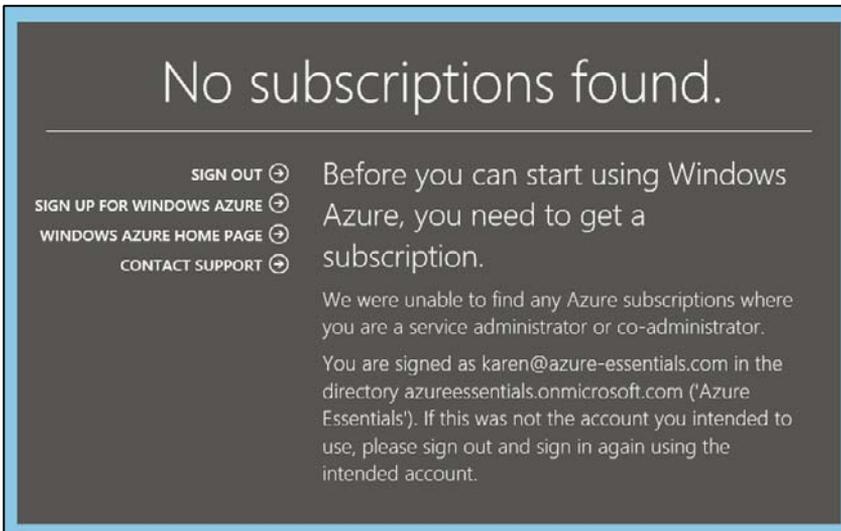


FIGURE 7-1 No subscriptions found.

See Also For more information on managing the directory for your Office 365 subscription in Azure, please see <http://msdn.microsoft.com/library/azure/dn629580.aspx>.

Note As of the time of this writing, Azure Active Directory is available only in the Azure Management Portal (<http://manage.windowsazure.com>).

If you don't yet have a subscription to a Microsoft cloud service such as Azure or Office 365, the act of signing up for the service will automatically create an Azure AD directory. You cannot use Azure without a directory. If you signed up for Azure prior to July 7, 2013, you likely were automatically assigned a Default directory. You can associate your subscription with a different Azure AD directory by using the Azure Management Portal. Proceed to the SETTINGS extension in the left navigation area, select your subscription, and then click EDIT DIRECTORY on the bottom command bar. The resulting

dialog box, shown in Figure 7-2, will allow you to select a different Azure AD directory to be associated with the selected Azure subscription.

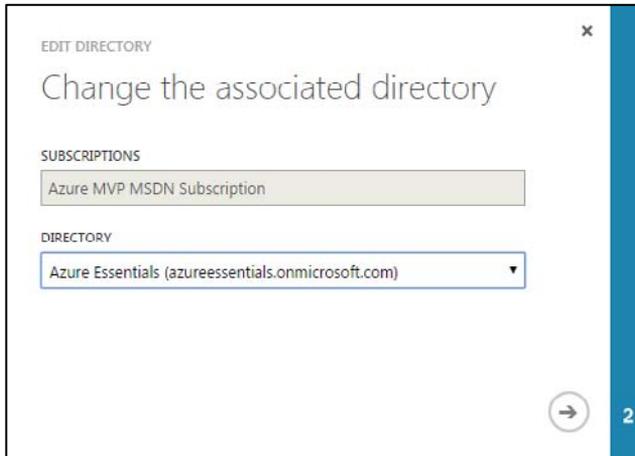


FIGURE 7-2 Change the Azure AD directory associated with an Azure subscription.

You can add a new Azure AD directory from the Azure Management Portal. To do so, select the +NEW button on the bottom command bar and then navigate to APP SERVICE, ACTIVE DIRECTORY, DIRECTORY, and finally CUSTOM CREATE, as seen in Figure 7-3.

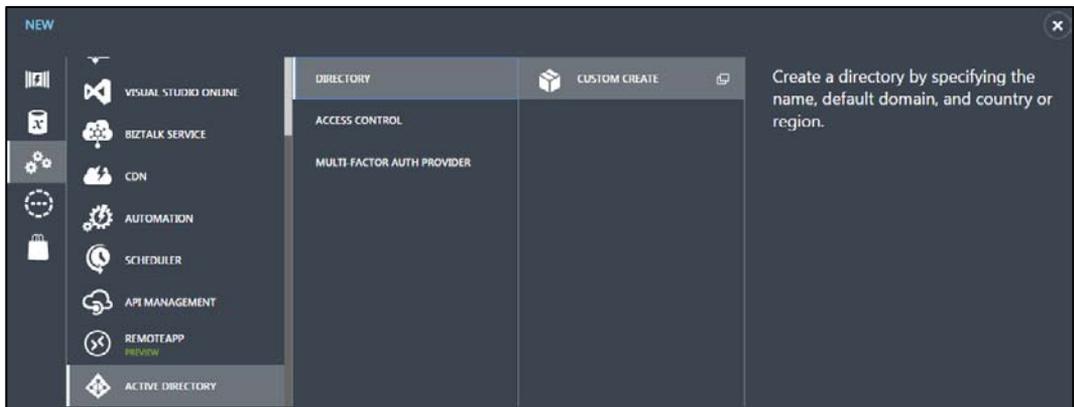


FIGURE 7-3 Create a new Azure AD directory.

In the resulting Add Directory dialog box, provide a friendly name for the directory, provide a unique domain name, and select your country or region, as shown in Figure 7-4.

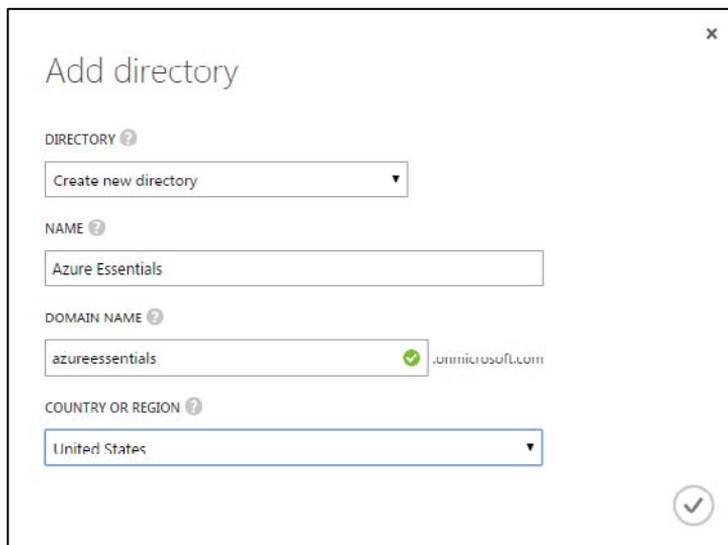


FIGURE 7-4 Add a new Azure AD directory.

Custom domains

Notice the domain name associated with the directory: `[directory_name].onmicrosoft.com`. Every Azure AD directory gets a unique name associated with the `*.onmicrosoft.com` domain. As a result, every user in the directory would have a name such as `mike@azureessentials.onmicrosoft.com`.

You are not forced to always use the `*.onmicrosoft.com` domain name. Instead, you can assign a custom domain, one that you own. Once a custom domain is established, users in the directory would reference the custom domain name instead (for example, `mike@azure-essentials.com`).

The process to associate a custom domain with your Azure AD directory is relatively easy. The Azure AD section of the Azure Management Portal will walk you through all the steps in an easy-to-follow wizard. There are three basic steps:

1. Get some basic information about your domain (or obtain a new domain if needed).
2. Create a DNS record to prove ownership of the domain.
3. Verify the domain.

First, select the desired directory in the Azure AD section of the Azure Management Portal and then click DOMAINS in the top navigation section. As seen in Figure 7-5, if you don't already have a custom domain, the portal will prompt you to create a custom domain and provide an ADD A CUSTOM DOMAIN link to get started.

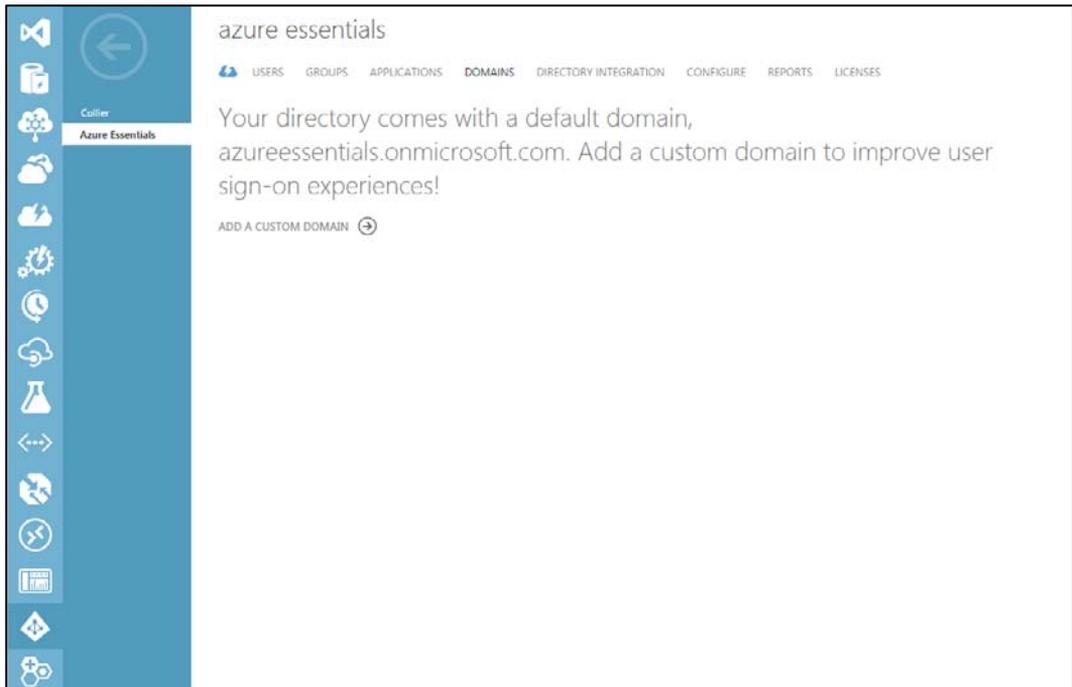


FIGURE 7-5 Add a custom domain.

After you click the ADD A CUSTOM DOMAIN link, a dialog box will open, as shown in Figure 7-6, prompting you to add the desired domain name to the list of potential domain names associated with your Azure AD directory.

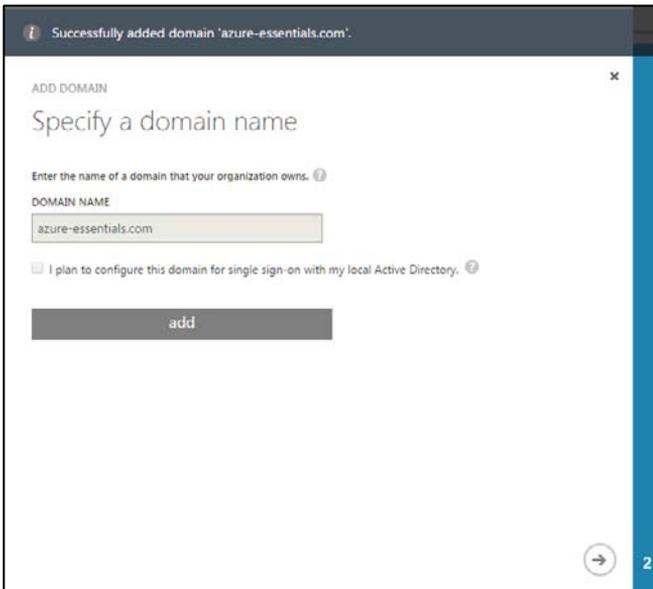


FIGURE 7-6 Specify a domain name.

The next step will be to prove that you own the domain name. Step 2 of the wizard, shown in Figure 7-7, prompts you to add a DNS setting (TXT or MX record) at your domain name registrar. For the purposes of this example, GoDaddy is used; however, the wizard provides a link to an MSDN article that provides detailed steps for several popular registrars.

ADD DOMAIN ×

Verify azure-essentials.com

Go to your domain name registrar and update the DNS settings for azure-essentials.com.
[Instructions for adding a DNS record at popular domain name registrars](#)

Add the record type that is supported by your domain name registrar for azure-essentials.com.

RECORD TYPE	<input type="text" value="TXT record"/>
ALIAS OR HOST NAME	<input type="text" value="@"/>
DESTINATION OR POINTS TO ADDRESS	<input]"="" type="text" value="MS=m:"/>
TTL	<input type="text" value="1 Hour"/>

1

FIGURE 7-7 Verify domain settings.

For GoDaddy, proceed to the DNS Manager section where you can work with DNS records. There you will need to add a new TXT record (if that was the record type selected in the Azure Management Portal). As shown in Figure 7-8, provide the Host, TXT Value, and TTL settings that were also specified in the Azure Management Portal wizard.

The image shows a dialog box titled "Add Zone Record" with a close button (X) in the top right corner. The domain "AZURE-ESSENTIALS.COM" is displayed at the top. Below the domain, there are four input fields:

- Record type:** A dropdown menu with "TXT (Text)" selected.
- Host:** A text input field containing "@".
- TXT Value:** A text input field containing "MS=m" followed by a redacted area.
- TTL:** A dropdown menu with "1 Hour" selected.

At the bottom of the dialog, there are three buttons: "Add Another" (green), "Finish" (green), and "Cancel" (blue).

FIGURE 7-8 Add Zone Record dialog box.

Click Finish in the GoDaddy editor and then be sure to save the zone file. After adding the TXT record and saving the zone file, you can attempt to verify the domain from the Azure Management Portal wizard by clicking VERIFY. If the verification is successful, you will receive a notification, as seen in Figure 7-9.

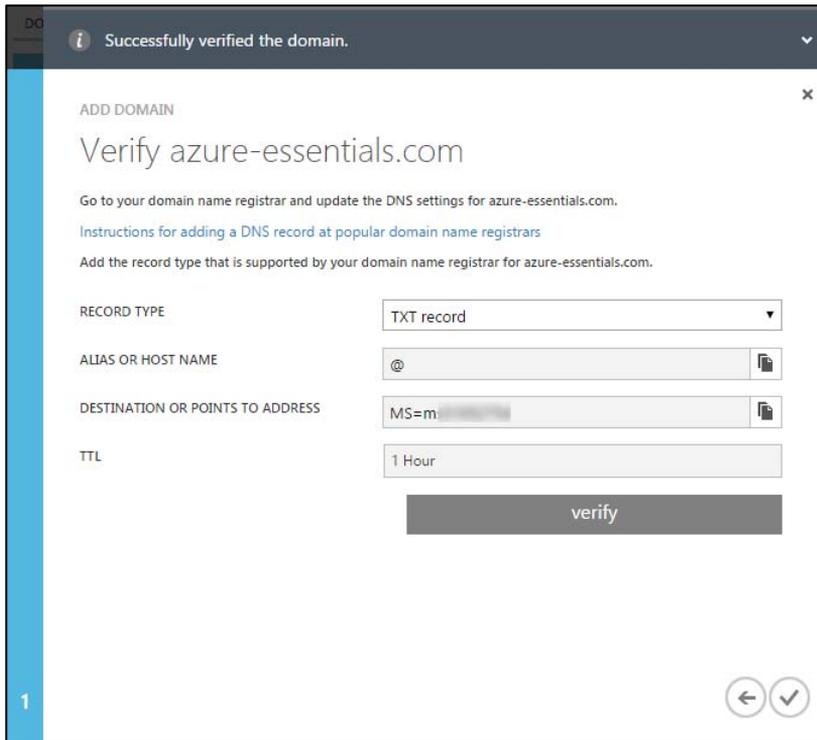


FIGURE 7-9 Successfully verifying the domain.

It could take 15 minutes for the DNS changes to take effect. In some cases, it might take up to 72 hours for the DNS records to propagate. If you are unable to verify the domain after 72 hours, you should return to the domain registrar site to verify the DNS record information is correct.

After exiting the wizard, you should notice both the custom domain and the default (or basic) domain listed in the DOMAINS section, as shown in Figure 7-10. If you want to add more custom domains, repeat the steps by first clicking the ADD button in the bottom command bar. Adding more custom, verified domains will allow you to change the primary domain associated with the default *.onmicrosoft.com domain.

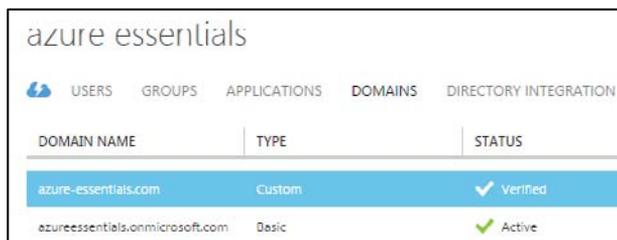


FIGURE 7-10 Verified custom domain.

Delete a directory

It is possible to create (or be associated with, including being a member of) a maximum of 20 directories. Creating multiple Azure AD directories can be helpful in development and testing scenarios in which you might not have access to the production subscription. When finished with a directory, you can easily delete that directory. Because deleting a directory is a potentially significant action, a few conditions (enforced by Azure) must be met to delete a directory:

- No users in the directory other than the global administrator
- No applications in the directory
- No subscriptions (for example, Azure, Office 365, and so on) associated with the directory
- No Multi-Factor Authentication providers linked to the directory

To delete the directory, just select the directory and click DELETE on the bottom command bar.

Users and groups

After creating the Azure AD directory, a common next step is to add users to the directory. Once users are in the directory, those users will be able to take advantage of Azure AD features such as SSO, access to the application gallery, and Multi-Factor Authentication, more details of which are provided later in this chapter.

Add users

Users in Azure AD can be one of three types:

- **A user in your organization** The user is created and managed in your directory.
- **A user with a Microsoft account (for example, hotmail.com or outlook.com)** This is often done as a way to collaborate on Azure resources by granting the user co-administrative rights to the Azure subscription.
- **A user in another Azure AD directory** The user is sourced from another Azure AD directory.

To add new users to the directory, there are a few different approaches you can take:

- Users can be in the directory as a result of the directory originating with a Microsoft cloud service such as Office 365.
- Users can be synchronized from an on-premises Windows Server Active Directory instance by using AADSync.
- Users can be added programmatically via PowerShell or the Azure AD Graph REST API.
- Users can be added manually via the Azure Management Portal.

When you first create a new Azure AD directory, and you are using an Azure subscription associated with a Microsoft account, there will already be one user account in the directory: yours. You can then add users to the directory by going to the USERS section in the desired directory and clicking the ADD USER button on the bottom command bar, as shown in Figure 7-11.

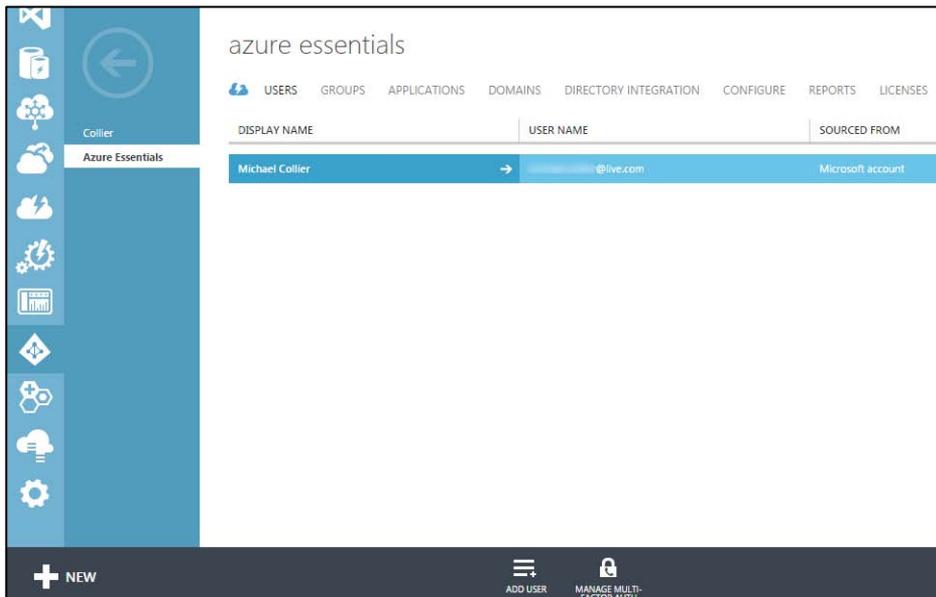


FIGURE 7-11 Listing of users in the Azure AD directory.

Clicking the ADD USER button opens a new dialog box that allows you to provide details for the new user. As seen in Figure 7-12, if the user is in your organization, you can select the domain name suffix for that user, either the default *.onmicrosoft.com or a custom domain (if configured).

ADD USER

Tell us about this user

TYPE OF USER

New user in your organization

USER NAME ?

sonja

azureessentials.onmicrosoft.com

azure-essentials.com

azureessentials.onmicrosoft.com

2 3

FIGURE 7-12 Add a user in your organization.

Figure 7-13 shows the next step, in which you will be able to provide more information about the user, such as the user's first and last name, display name, and potentially an administrative role for the user. You can also enable Multi-Factor Authentication for the user. (More information on that is provided later in this chapter.)

ADD USER

user profile

FIRST NAME: Sonja

LAST NAME: Collier

DISPLAY NAME: Sonja Collier

ROLE: User

MULTI-FACTOR AUTHENTICATION: Enable Multi-Factor Authentication

FIGURE 7-13 New user profile.

See Also For more details on the administrative roles, please reference <http://msdn.microsoft.com/library/azure/dn468213.aspx>.

The final step in creating a new user in your organization is to get a temporary password for the user. Clicking the CREATE button, as shown in Figure 7-14, creates and assigns a temporary password for the newly created user.

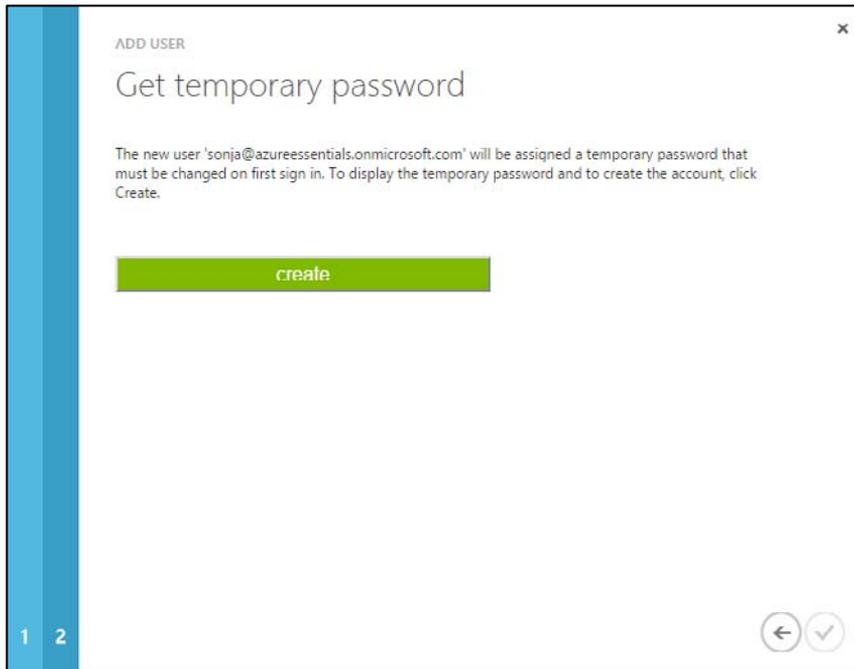


FIGURE 7-14 Get temporary password.

After creating and assigning the temporary password, the password is displayed and you will have an opportunity to have the password sent via email, as seen in Figure 7-15.

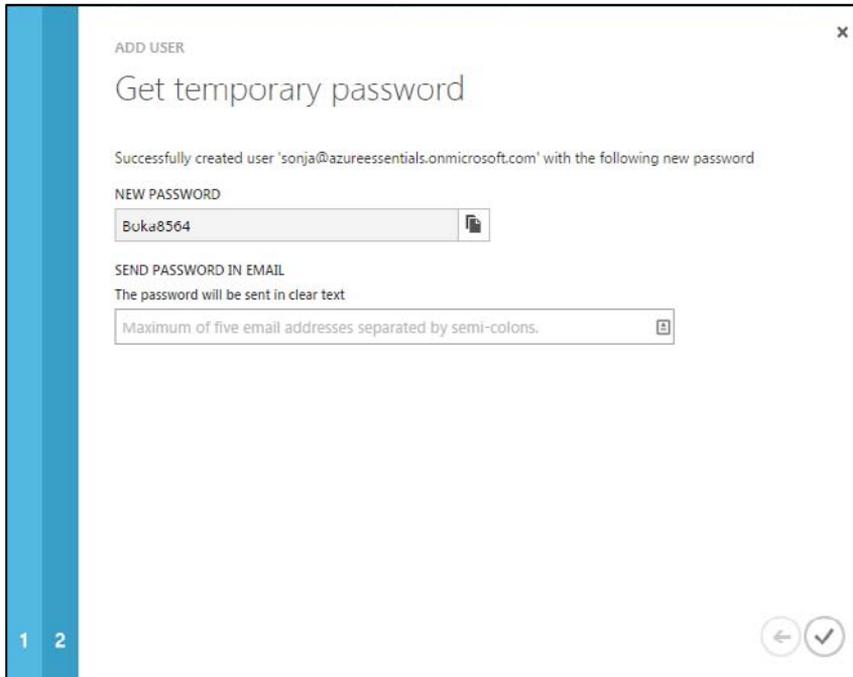
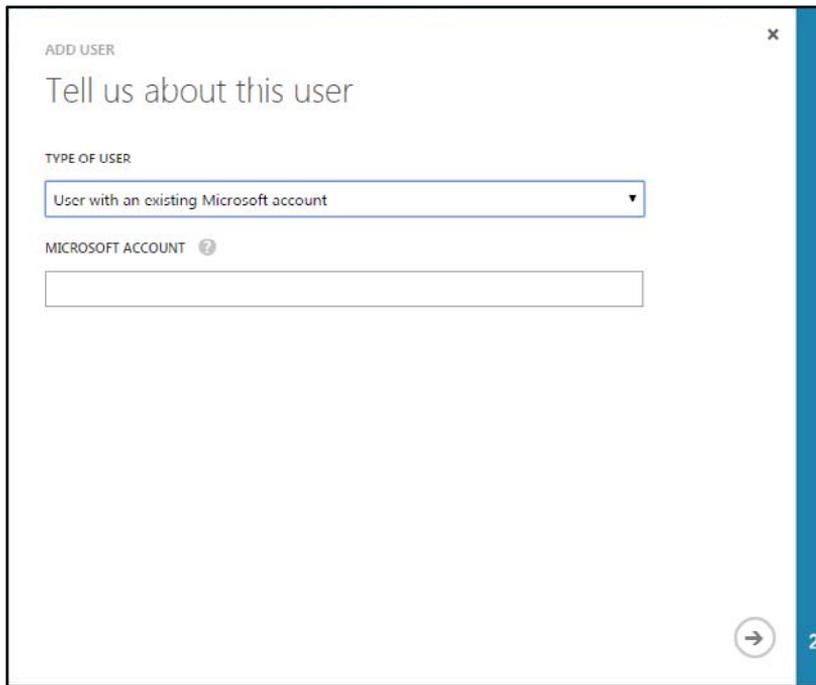


FIGURE 7-15 Display and optionally send the temporary password.

The process for adding a new user with an existing Microsoft account is similar. Instead of assigning the user to a domain name associated with your organization, you will use the user's Microsoft account (email address), as shown in Figure 7-16.

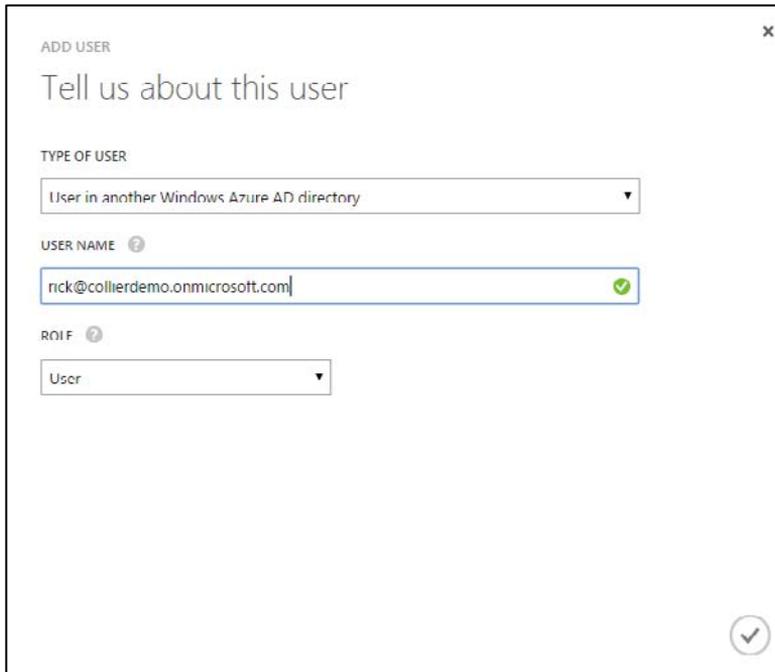


The screenshot shows a dialog box titled "ADD USER" with a close button (X) in the top right corner. The main heading is "Tell us about this user". Below this, there is a section labeled "TYPE OF USER" with a dropdown menu currently showing "User with an existing Microsoft account". Underneath is a section labeled "MICROSOFT ACCOUNT" with a help icon (?) and an empty text input field. At the bottom right of the dialog, there is a circular button with a right-pointing arrow and the number "2" next to it.

FIGURE 7-16 Add a user with an existing Microsoft account.

After providing the user's email address, you will be able to provide the user's first name, last name, display name, and role, just like when adding a user in your organization.

Finally, when adding a new (external) user who is a user in another Azure AD directory, you will need to provide the name for the user. This is done in the form of the user's email account (or in Active Directory terms, the user's User Principal Name, or UPN), as seen in Figure 7-17. Keep in mind that to select a user in another Azure AD directory, you also will need to be an administrator in the other directory.



The screenshot shows a dialog box titled "ADD USER" with a close button (X) in the top right corner. The main heading is "Tell us about this user". Below this, there are three input fields:

- TYPE OF USER:** A dropdown menu with the selected option "User in another Windows Azure AD directory".
- USER NAME:** A text input field containing "rick@collierdemo.onmicrosoft.com" with a green checkmark icon to its right.
- ROLES:** A dropdown menu with the selected option "User".

A green checkmark icon is located in the bottom right corner of the dialog box.

FIGURE 7-17 Add an external user.

Adding the external user to your directory copies the display name and user name from the source or home directory to your directory. The user would still authenticate against his or her home directory, but any changes to user's properties (email address, display name, job title, and so on) do not propagate to your directory.

To determine where users in the directory originated, look at the SOURCED FROM column in the USERS section, as seen in Figure 7-18. If users were synchronized from a Windows Server Active Directory using AADSync, the SOURCED FROM column would indicate Local Active Directory. Similarly, users from Office 365 would have the value Office 365 in the SOURCED FROM column.

DISPLAY NAME	USER NAME	SOURCED FROM
Michael Collier	...@live.com	Microsoft account
Mike Collier	...@hotmail.com	Microsoft account
Rick Collier	rick@collierdemo.onmicrosoft.com	Windows Azure AD (other directory)
Robin Shahan	robin@azure-essentials.com	Windows Azure Active Directory
Sonja Collier	sonja@azureessentials.onmicrosoft.com	Windows Azure Active Directory

FIGURE 7-18 User list and source location.

Add groups

It is a common practice in Active Directory to organize users into groups. Groups make it easier to assign rights or grant access to resources. Instead of granting access for each individual user, access is granted to the group of which the user is a member. The user inherits the access rights of the group.

You also can create and manage groups in an Azure AD directory. Groups in Azure AD can be helpful when granting access to Software-as-a-Service (SaaS) applications available in the Azure AD application gallery (discussed later in this chapter). The GROUPS section of the selected Azure AD directory allows you to create and manage groups. If no groups exist, create a group by clicking the ADD GROUP button on the bottom command bar or the ADD A GROUP link, as shown in Figure 7-19.

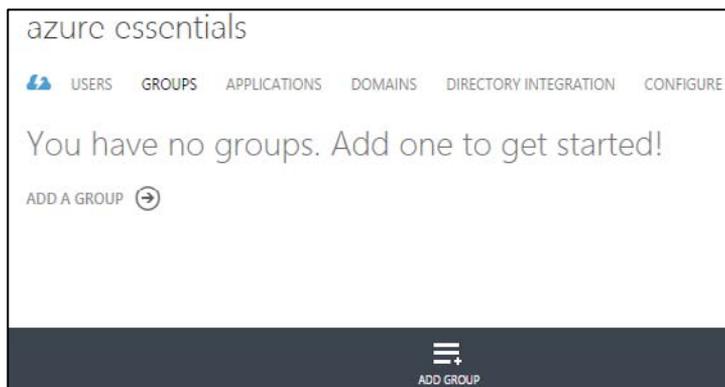


FIGURE 7-19 Add a group.

In the Add Group dialog box, you can provide a name and description for the new group, as seen in Figure 7-20.

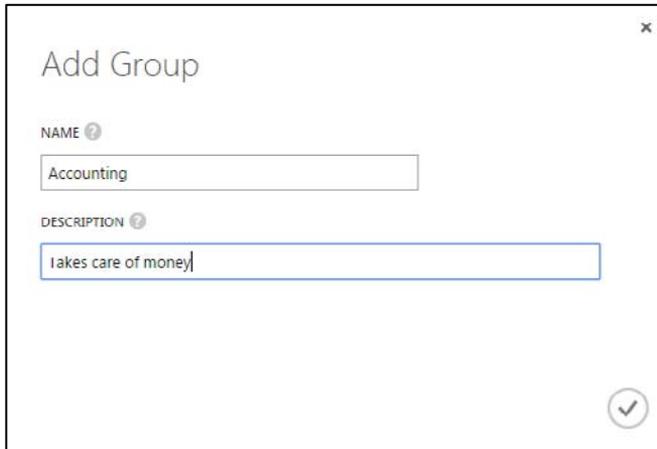


FIGURE 7-20 Add group details.

To add users to the group, select the group name to open a new screen that allows you to manage the group and then click the **ADD MEMBERS** link or the **ADD MEMBERS** button on the bottom command bar, as shown in Figure 7-21.

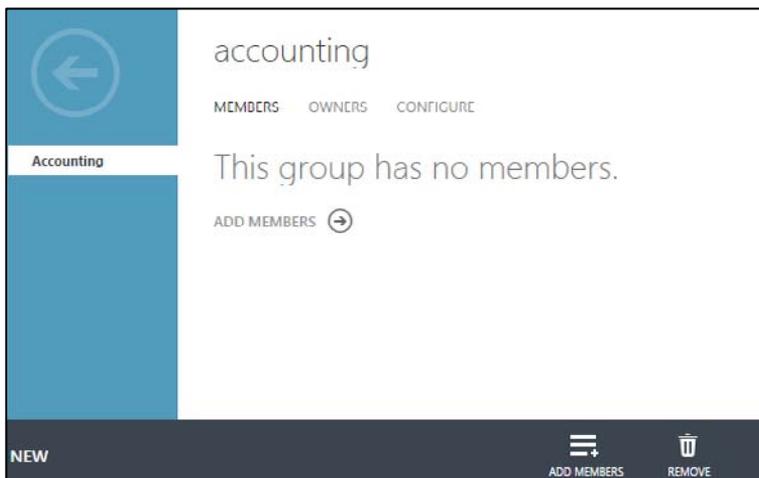


FIGURE 7-21 Group with no members.

As seen in Figure 7-22, the Add Members dialog box enables you to select users, or other groups in the current directory, to add as members to the group selected.

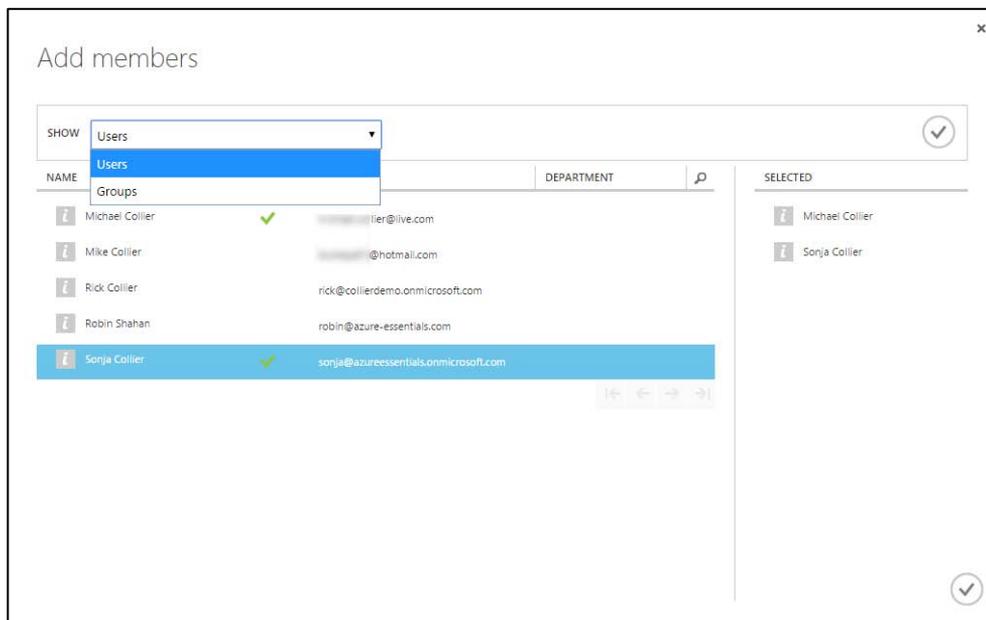


FIGURE 7-22 Add members to a group.

Multi-Factor Authentication

Azure Multi-Factor Authentication (MFA) provides additional security for on-premises or Azure-hosted solutions. For the purposes of this chapter, only Azure-hosted solutions are discussed. Azure MFA works by injecting a second authentication challenge that the user must successfully complete. Azure MFA complements password-based authentication challenge (something you know) with a challenge based on something you have: a phone call, text message, or mobile app notification. Having multiple layers of protection makes it harder for attackers to compromise and access the account.

MFA comes in a few different varieties. Azure AD Free and Azure AD Basic support MFA for Azure administrators, whereas Azure AD Premium adds MFA support for users. MFA for Azure administrators provides security for their administrative account, thus adding security related to creating and managing resources such as Azure Virtual Machines, Azure Websites, and so on. MFA for users provides additional security for users signing into applications configured to support Azure AD.

Azure Multi-Factor Authentication

To begin using the full capabilities of Azure MFA, you will first need to add a MFA Provider in Azure AD. To add a MFA Provider, go to the ACTIVE DIRECTORY section in the Azure Management Portal and then click the MULTI-FACTOR AUTH PROVIDER area. If no providers exist, you are presented with an option to create one, as seen in Figure 7-23.

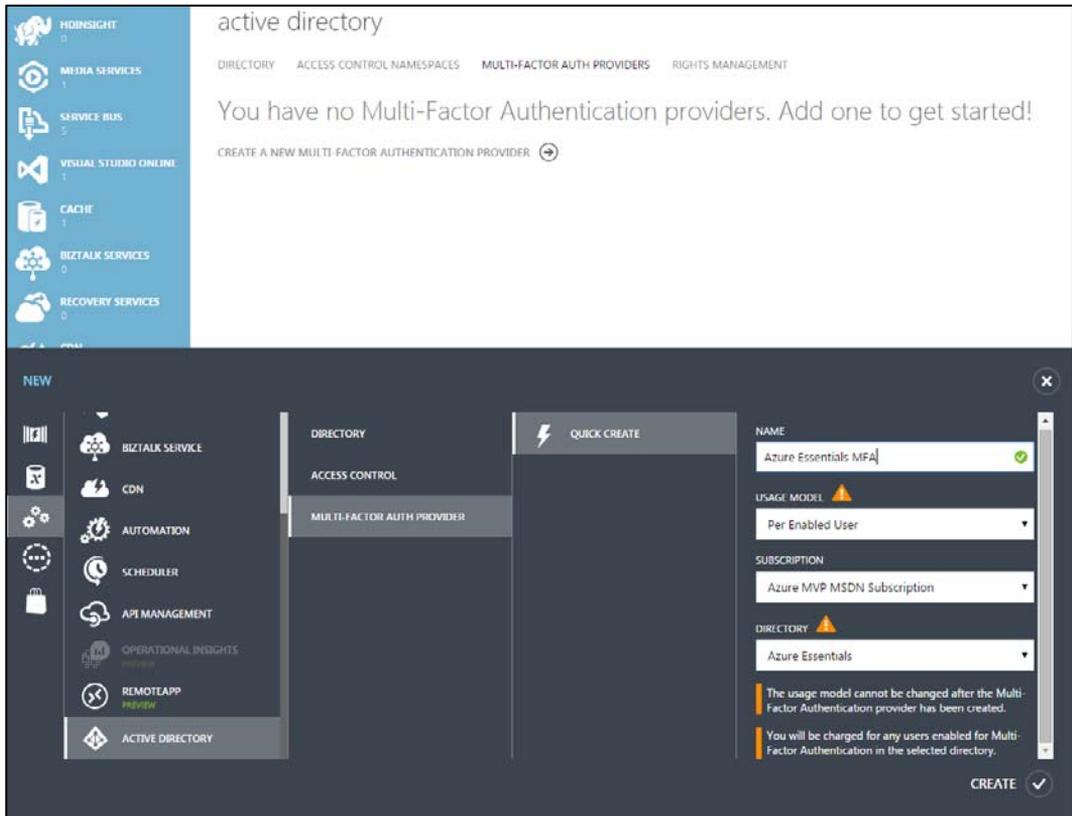


FIGURE 7-23 Create a new Azure Multi-Factor Authentication Provider.

After creating a new MFA provider, you can customize many advanced features of the service by clicking the MANAGE button, shown in Figure 7-24, to launch a new browser window that loads the Azure Multi-Factor Authentication management portal.

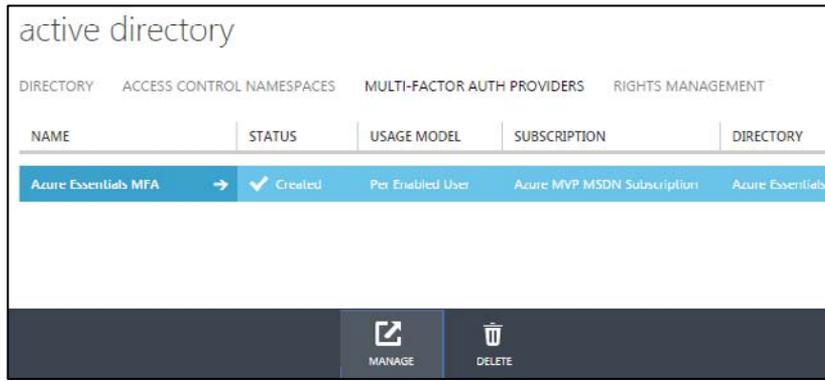


FIGURE 7-24 Azure MFA providers.

Note Azure Multi-Factor Authentication is a result of Microsoft's acquisition of PhoneFactor in October 2012. The Azure Multi-Factor Authentication management portal continues to reside on a phonefactor.net domain.

See Also For more information on Azure Multi-Factor Authentication, please refer to <http://azure.microsoft.com/en-us/documentation/services/multi-factor-authentication/>.

Multi-Factor Authentication for Azure administrators

Recall from earlier in this chapter that when creating a new user, you had the option to enable MFA. Selecting the user's name and then the MANAGE MULTI-FACTOR AUTH button on the command bar, as shown in Figure 7-25, launches a new browser window and loads a site allowing you to manage MFA settings.

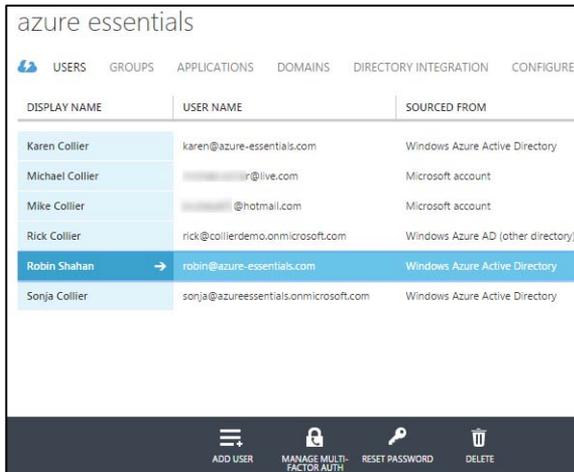


FIGURE 7-25 Manage Multi-Factor Authentication.

Alternatively, if the user is not enabled for MFA, you can enable it from this site by selecting the desired user(s) and clicking Enable, as shown in Figure 7-26.

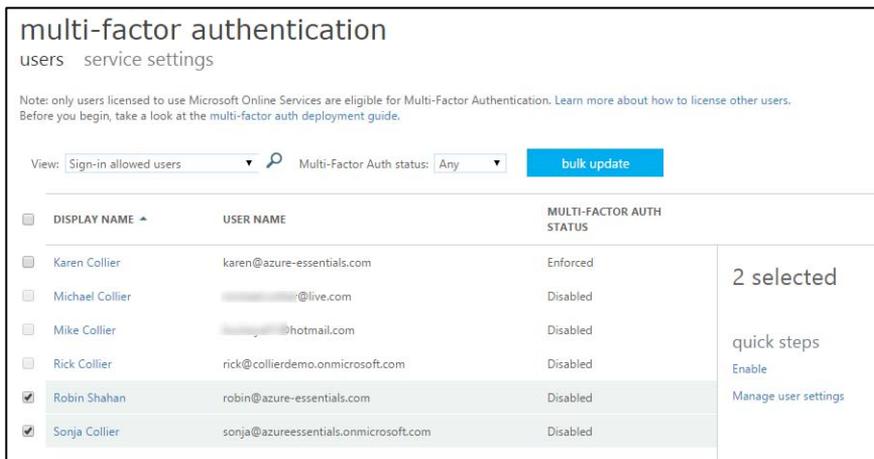


FIGURE 7-26 Enable Multi-Factor Authentication for multiple users.

If MFA is enabled for a user, and the user attempts to access the Azure management portal (the user must also be a co-administrator on the Azure subscription), he or she will be presented with an additional security verification step. As seen in Figure 7-27, the first time the user attempts to log in, the user will be asked to specify the desired contact method—phone, text message, or mobile application—and complete the necessary steps.

additional security verification

You are required to sign in with your password as well as a registered device. This makes it harder for a hacker to sign in with just a stolen password. Follow these steps to get your account set up. [View video](#)

Step 1: Specify the contact method we should use by default

Authentication phone

Mode

Send me a code by text message

Call me

[next](#)

Your phone numbers will only be used for account security. Standard telephone and SMS charges will apply.

©2014 Microsoft Legal | Privacy

FIGURE 7-27 Additional security verification details.

On subsequent login attempts, the user will need to provide the requested additional security verification answer. As shown in Figure 7-28, a text message verification code is requested.

Help us protect your account

We've sent you a text message with a verification code.

 karen@azure-essentials.co...
+X XXXXXXXX46

[Sign in](#)

[Use a different verification option](#)

[Sign out and sign in with another account](#)

[More information](#)

FIGURE 7-28 Text message verification code.

Application gallery

The Azure AD application gallery provides access to more than 2,400 (and growing) popular SaaS applications such as Box, DocuSign, Salesforce, Google Apps, and many more. Instead of IT administrators configuring access to each application separately, and potentially managing various disparate logins, Azure AD simplifies the process by enabling SSO for the applications.

Azure AD supports three options for SSO:

- **Azure AD Single Sign-On** Uses the user's account information directly from Azure AD. If the user is already signed into Azure AD (or likewise Office 365), there is no need for the user to reauthenticate when accessing the third-party SaaS application. A limited number of applications in the Azure AD application gallery support federation-based SSO.
- **Password Single Sign-On** Uses the user's account information from the third-party SaaS application. With this approach, the user's account information and password is collected, securely stored, and provided to the SaaS application via a web browser extension. The majority of applications in the Azure AD application gallery support this form of SSO. The browser extension is supported on Internet Explorer (IE8, IE9, and IE10 on Windows 7 or later) and Chrome (Windows 7 or later; MacOS X or later).
- **Existing Single Sign-On** Uses an existing Active Directory Federation Services (AD FS) or other third-party provider for SSO.

Adding gallery applications

To add gallery applications to your Azure AD directory, first select the desired directory and then the APPLICATIONS section from the top navigation options. If you don't have any applications in the gallery, click ADD AN APPLICATION to open a new dialog box, as shown in Figure 7-29, presenting you with two options: add an application you are developing or add an application from the gallery.

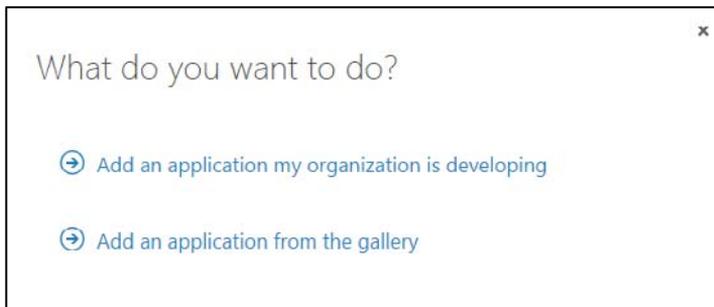


FIGURE 7-29 Add an application to Azure AD.

If you, or another member of your organization, have developed an application that would like to take advantage of Azure AD features (for example, SSO, ability to query the Azure AD Graph API, and so on), then select the first option.

Selecting the second option will allow you to register third-party SaaS applications with your Azure AD directory. As can be seen in Figure 7-30, there are many applications available. Use the filter box in the upper-right corner to search by name for a specific application.

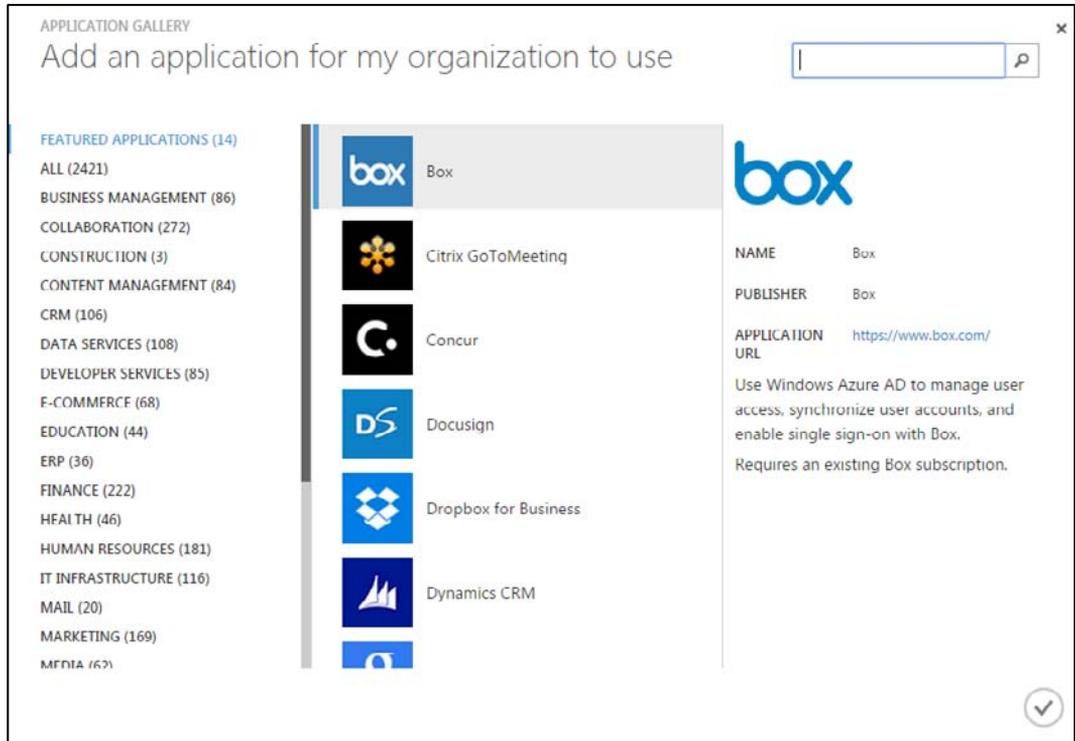


FIGURE 7-30 Select a gallery application.

After you select the desired application(s), each application will appear on the list in the APPLICATIONS section. As shown in Figure 7-31, you can add an application by clicking ADD, and you can remove an application by selecting the application and clicking DELETE.

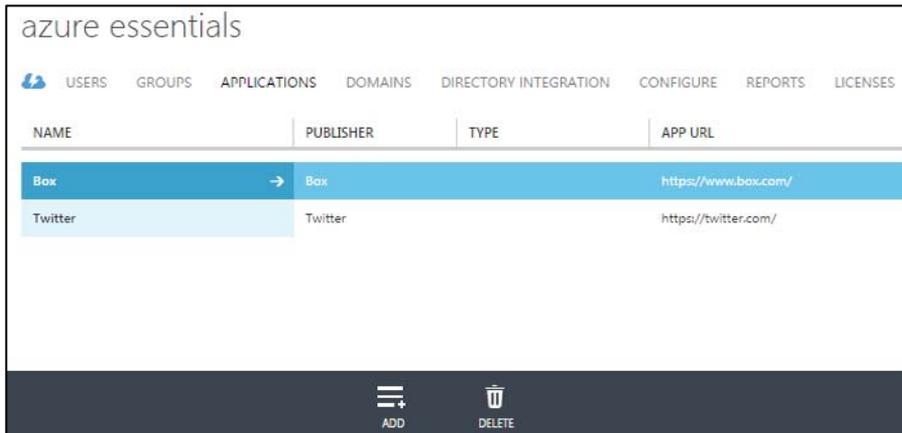


FIGURE 7-31 List, add, and delete applications.

Assigning users to applications

As mentioned previously, there are different ways applications can leverage Azure AD for SSO. For example, Box provides all three options (Azure AD Single Sign-On, Password Single Sign-On, and Existing Single Sign-On), and Twitter provides only Password Single Sign-On and Existing Single Sign-On. Follow the on-screen guidance, as shown in Figure 7-32, to configure SSO for the selected application.

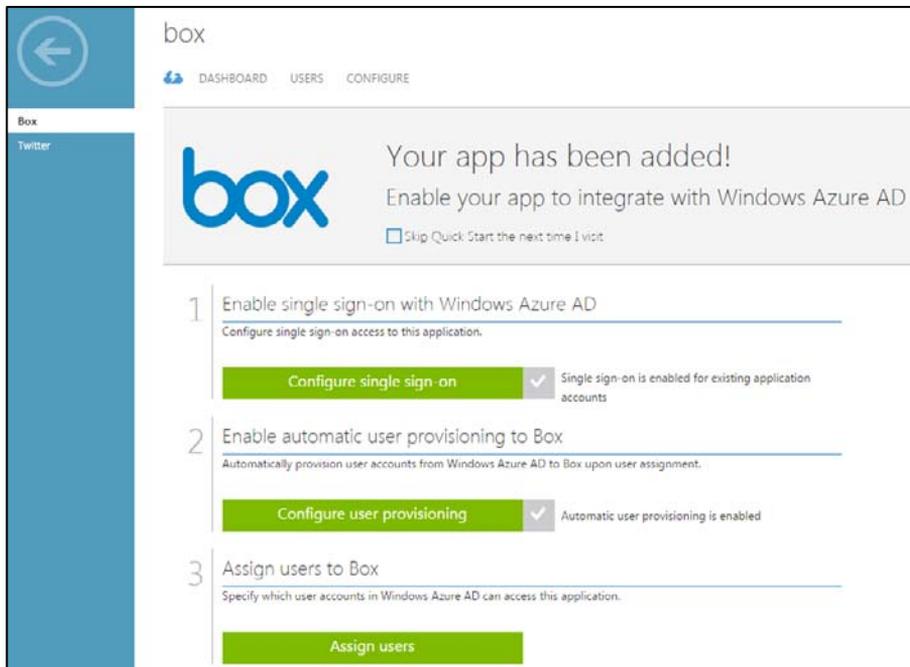


FIGURE 7-32 Configuration for gallery application.

Some applications, such as Box, have the ability to provision (create) users automatically into the application once the user is assigned access in Azure AD. If a user is deleted from Azure AD or his or her access is revoked, this change is automatically propagated to the SaaS application. This can greatly simplify IT administration responsibilities.

To grant Azure AD directory users access to the selected application, first click Assign Users to access a new screen allowing you to assign or revoke access. As shown in Figure 7-33, select the desired user and then click either ASSIGN or REMOVE on the bottom command bar.

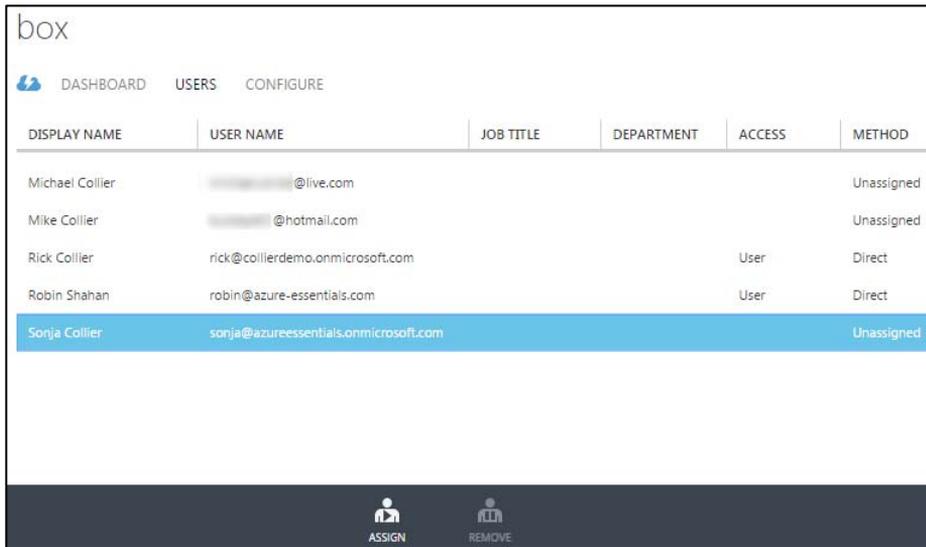


FIGURE 7-33 Assign users.

For applications that use Password Single Sign-On, after assigning users you will be presented with an option to allow users to enter and update their own credentials for the application, or you can enter them on their behalf, as shown in Figure 7-34.

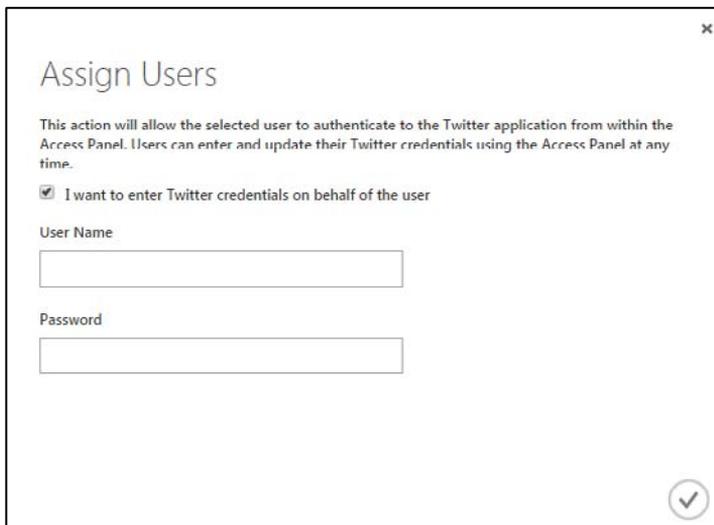


FIGURE 7-34 Assign user credentials.

See Also For detailed implementation details on integrating with and accessing the Azure AD application gallery, please refer to <http://msdn.microsoft.com/en-us/library/azure/dn308590.aspx>.

MyApps

Once users are assigned access to the SaaS applications available via the Azure AD application gallery, they can access those applications from the MyApps for Azure Active Directory site available at <http://myapps.microsoft.com>. Users will need to first authenticate with their Azure AD credentials (either *.onmicrosoft.com or associated custom domain name) to gain access to the site. If the user is already signed into a Microsoft cloud service site (Azure, Office 365, and so on), he or she will automatically be signed into the MyApps site. As can be seen in Figure 7-35, there are two main sections of the site: Applications and Profile.



FIGURE 7-35 Applications in MyApps site.

The Applications section provides a tile for each application to which the user has been granted access. If password-based SSO was selected as the authentication method when the application was added to the organization's Azure AD directory, the user might be prompted to install a browser component the first time the application is accessed.

To access the desired application, the user just needs to click the application's tile. If password-based SSO was selected, the user will be prompted to enter his or her credentials for the SaaS application, as shown in Figure 7-36 (unless the administrator who added the application did that on the user's behalf). The user is then redirected to the application and signed in using the appropriate credentials.

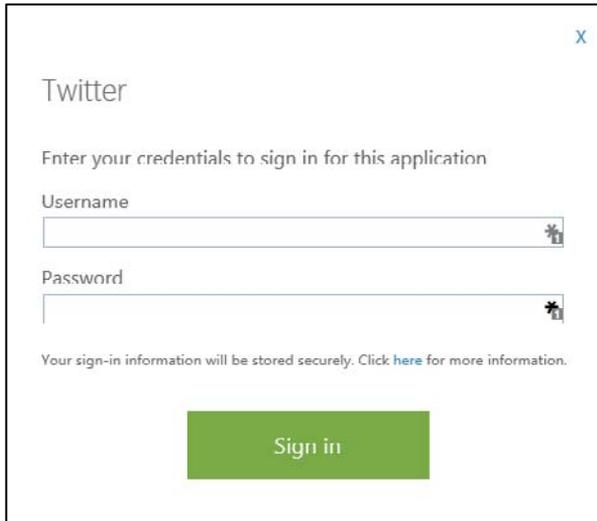


FIGURE 7-36 Provide user credentials for the application.

The Profile section, shown in Figure 7-37, provides some basic details about the current user and a way to change his or her password.

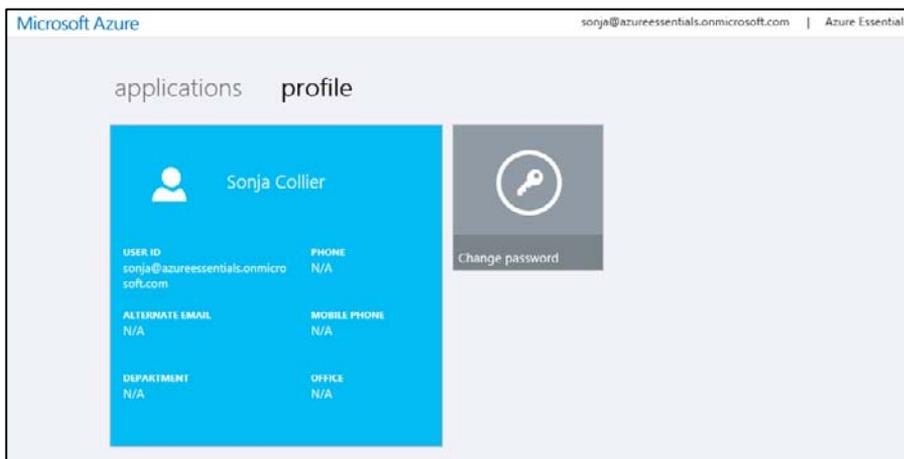


FIGURE 7-37 User profile in MyApps site.

Chapter 8

Management Tools

Throughout earlier chapters in this book, you have explored several prominent features of the Microsoft Azure platform. You have learned how to create Azure websites, cloud services, virtual machines, storage accounts, Azure SQL databases, and much more. The majority of examples have demonstrated using either the Azure Management Portal or the Azure Preview Portal. Although the portal(s) is a great way to work with Azure resources, other tools will also prove useful during development and management of those resources.

Management tool overview

There are many excellent tools available to aid in the development or management of Azure solutions—in fact, too many to cover in this chapter. Microsoft Visual Studio, PowerShell, and the Azure Cross-Platform Command-Line Interface (xplat-cli) tools are covered in this chapter. However, to assist in your awareness, Table 8-1 lists a few Azure features and related tools.

TABLE 8-1 Various tools for Azure management

Azure Feature	Tool Options
Azure SQL Database	SQL Server Management Studio
Azure Virtual Machines	<ul style="list-style-type: none">• Microsoft System Center• Puppet• Chef
Azure Storage	<ul style="list-style-type: none">• See the Azure Storage team blog post at http://blogs.msdn.com/b/windowsazurestorage/archive/2014/03/11/windows-azure-storage-explorers-2014.aspx for a list of popular tools.• AzCopy (included in the Azure SDK)
Azure Service Bus	Service Bus Explorer
Azure Management APIs	Microsoft Azure Management Library

For developers, Visual Studio provides a rich, integrated experience to develop, deploy, and maintain applications in Azure. For IT professionals and operations, the Azure PowerShell cmdlets and xplat-cli provide a robust and powerful scripting environment to manage resources deployed in Azure. In fact, some advanced features are only available via PowerShell. The xplat-cli tools provide a simple yet powerful means to manage Azure resources regardless of your operating system, because the xplat-cli works equally well across Windows, Linux, and Mac systems.

Visual Studio 2013 and the Azure SDK

Visual Studio is likely to be an Azure developer's primary interface for the development and management of Azure resources. This is especially the case for those developers creating solutions on the Microsoft technology stack (Windows, .NET, and so on). Developers using Linux or Mac systems will focus primarily on the xplat-cli. Any developer or IT professional can use either of the Azure management portals.

Install the Azure SDK

As an Azure developer, one of the first things you will want to do (after installing Visual Studio) is install the Microsoft Azure SDK for .NET. You can obtain the Azure SDK from the Azure Downloads page at <http://azure.microsoft.com/en-us/downloads/>, as shown in Figure 8-1. Install the SDK appropriate for your version of Visual Studio.

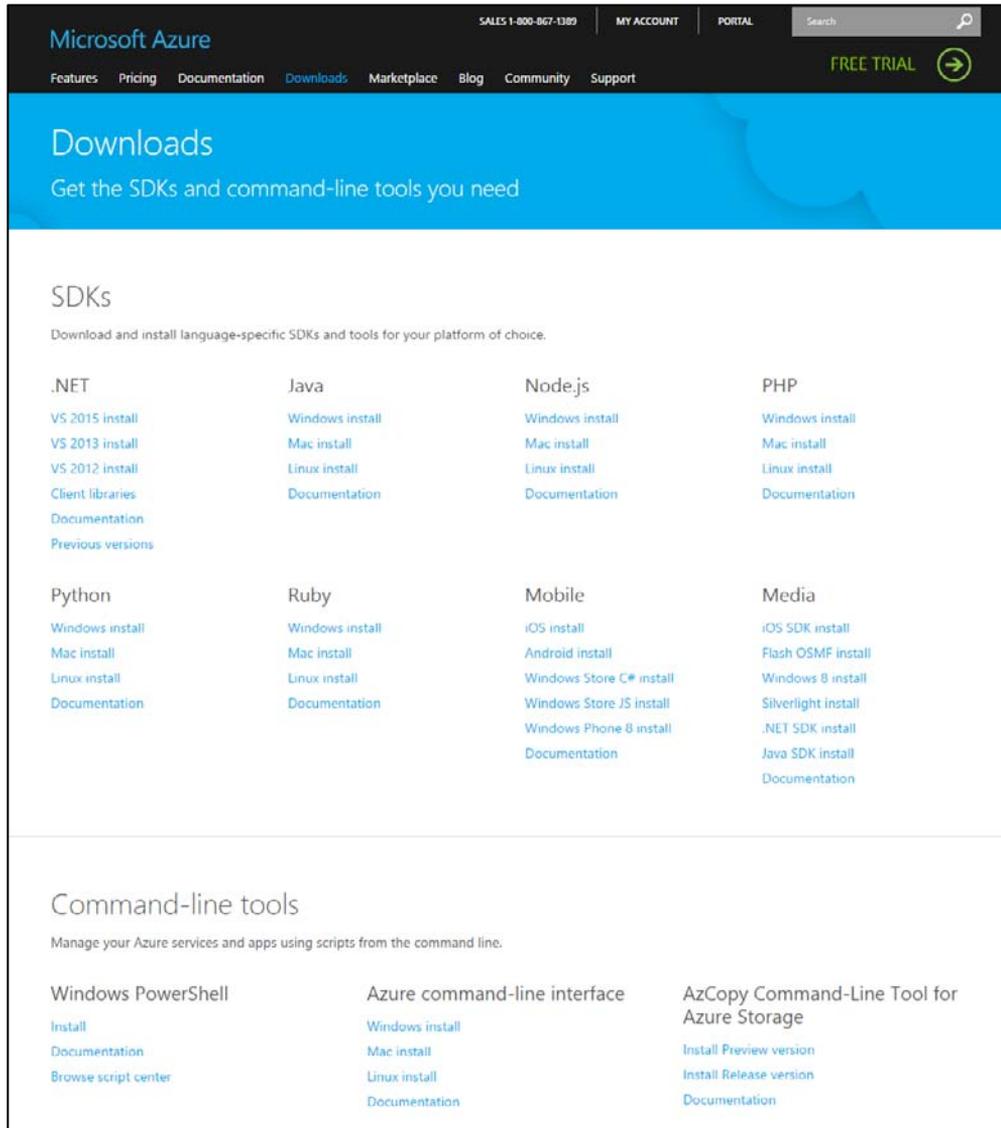


FIGURE 8-1 Azure Downloads page.

You will likely want to bookmark this page, as you will come back to it often because it is the source for various language SDKs (Java, PHP, Ruby, and so on) and the PowerShell and Azure command-line interface tools.

Installing the Microsoft Azure SDK for .NET is done using the Web Platform Installer (Web PI), as shown in Figure 8-2. The SDK includes any necessary client libraries required to work with Azure services such as Storage, Service Bus, Cloud Services, and so on. The SDK installation will also configure Visual Studio for full development and debugging support in Azure Websites and Cloud Services. Additionally, the SDK includes the Storage and Compute emulators—great for developing projects when you cannot be connected to the Internet (and thus Azure). Web PI also will take care of installing any necessary dependencies. The entire process will take several minutes.

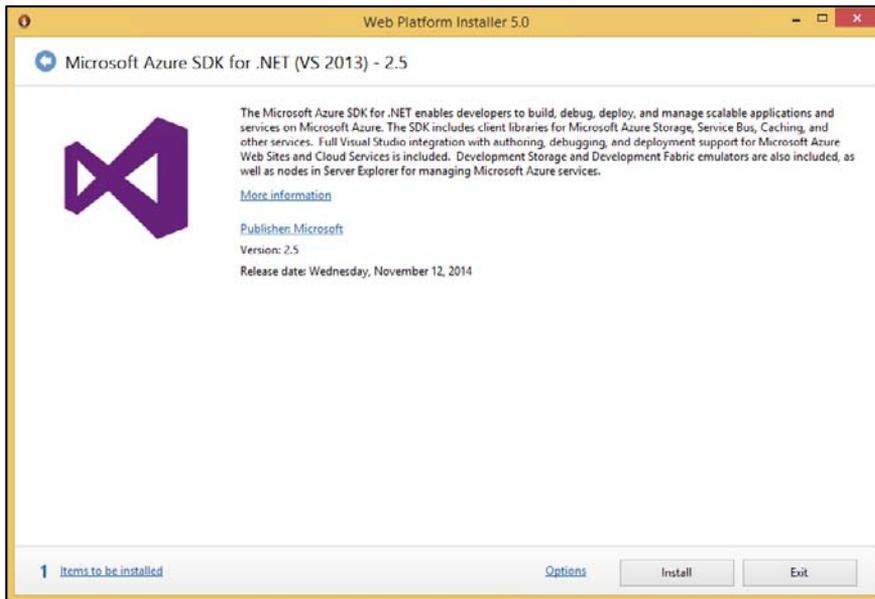


FIGURE 8-2 Microsoft Azure SDK for .NET (Visual Studio 2013).

When finished, you should see a dialog box like that shown in Figure 8-3, showing the various products installed as part of Microsoft Azure SDK for .NET.

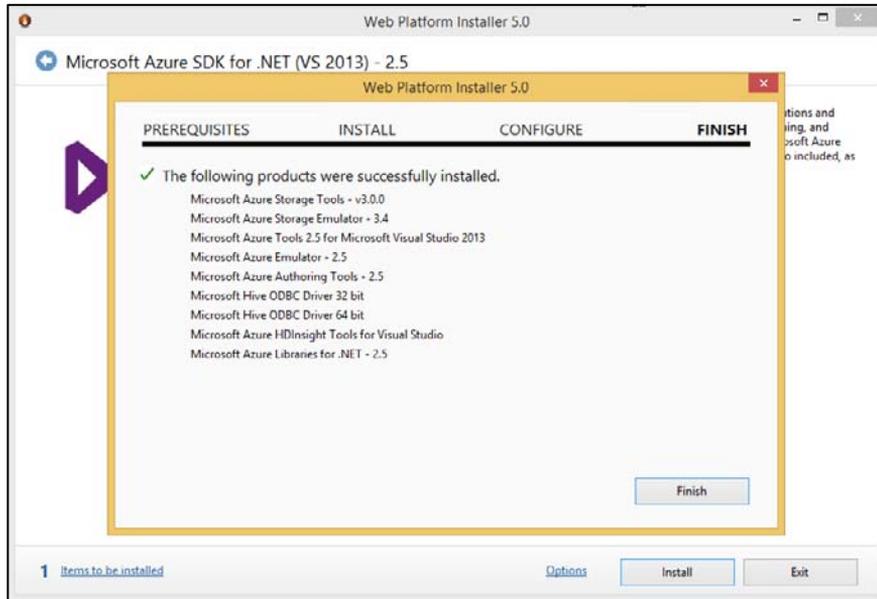


FIGURE 8-3 Completed Azure SDK installation.

See Also For release notes on the various Azure SDK for .NET versions, please refer to <http://msdn.microsoft.com/library/azure/dn627519.aspx>.

Manage resources with Server Explorer

After installing the Azure SDK, proceed to Server Explorer, shown in Figure 8-4. Server Explorer will have a new node, Azure. From the Azure node in Server Explorer, you can manage various Azure resources.

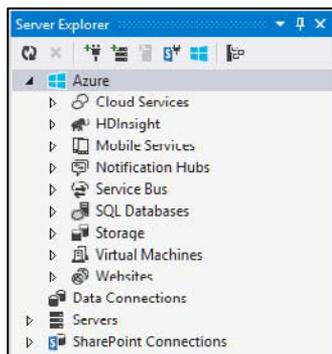


FIGURE 8-4 Visual Studio Server Explorer with Azure node.

The first thing you need to do is connect to your Azure subscription(s), which you can do by right-clicking the Azure node and selecting Connect To Microsoft Azure Subscription. This opens a dialog box, as seen in Figure 8-5, prompting for the email address of the account used to sign into Azure. This should be the same email address used when signing into either of the Azure management portals.

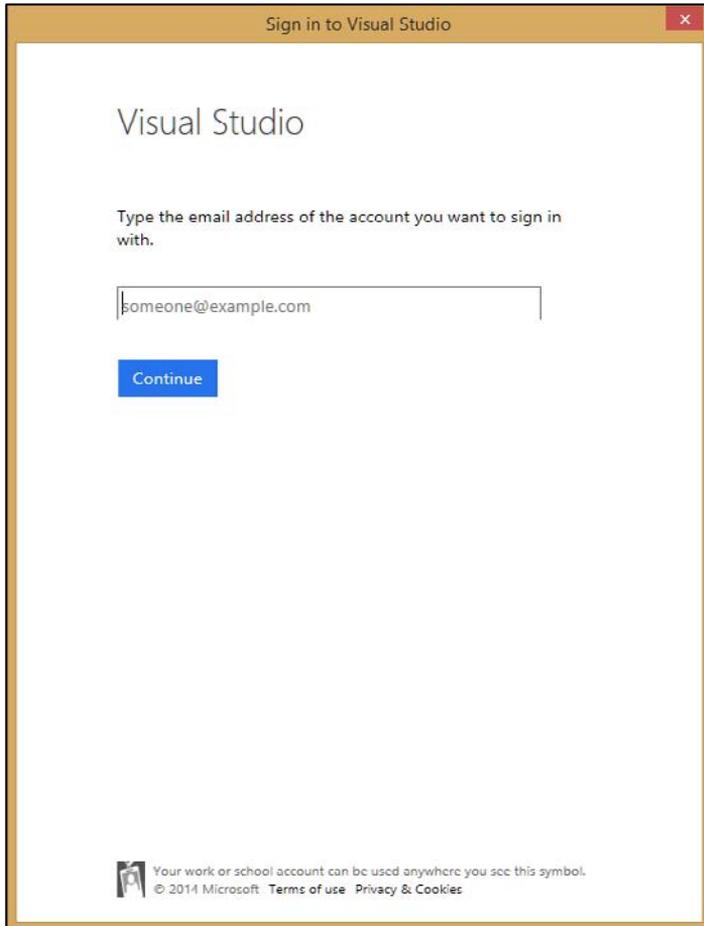


FIGURE 8-5 Sign into your Azure subscription.

After authenticating, Visual Studio will automatically download all the necessary configuration details to connect to any Azure accounts for which the provided email address is an administrator or co-administrator.

If you right-click the Azure node and select Manage Subscriptions, a Manage Microsoft Azure Subscriptions dialog box opens, as shown in Figure 8-6. In this dialog box, you can view all the subscriptions and Azure regions to which you have access using Visual Studio. If you want to filter out certain subscriptions or regions so that you do not see them while working with Azure resources in Visual Studio, just clear the appropriate check boxes.

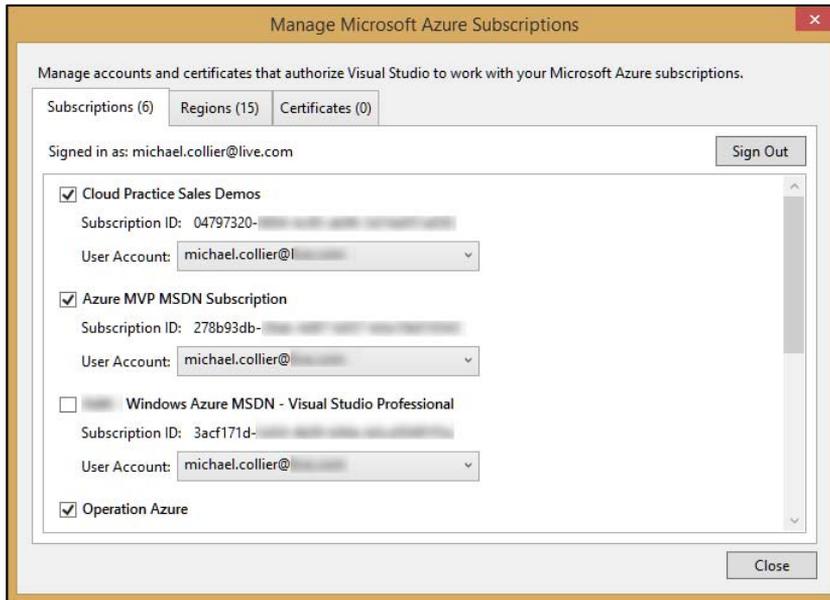


FIGURE 8-6 Manage Microsoft Azure Subscriptions.

To manage an Azure resource, just expand the node for that resource. Expanding the node lists all the resources under the selected type, as seen in Figure 8-7.

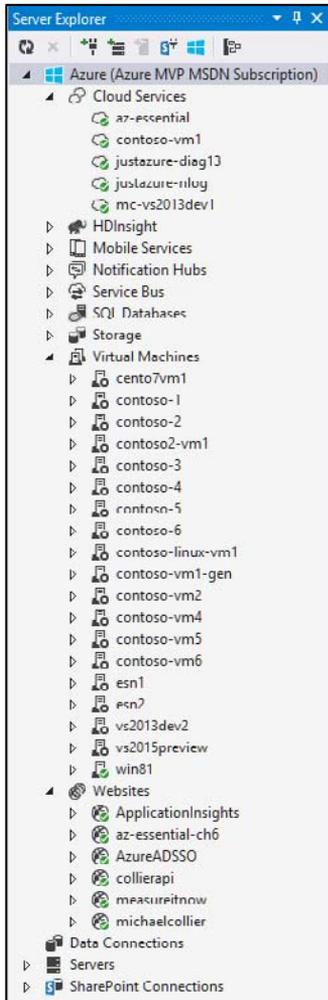


FIGURE 8-7 Expanding Azure resources.

Depending on the resource, selecting and expanding a node will list more details about the resource. For example, if you expand Virtual Machines, you see the public endpoints; if you expand Websites, you see information about the files and any related WebJobs. You also can right-click a node to get more contextual information.

For example, as shown in Figure 8-8, right-clicking a specific Websites node will provide contextual information to view the site in a browser, attach a debugger, view streaming logs, and more.

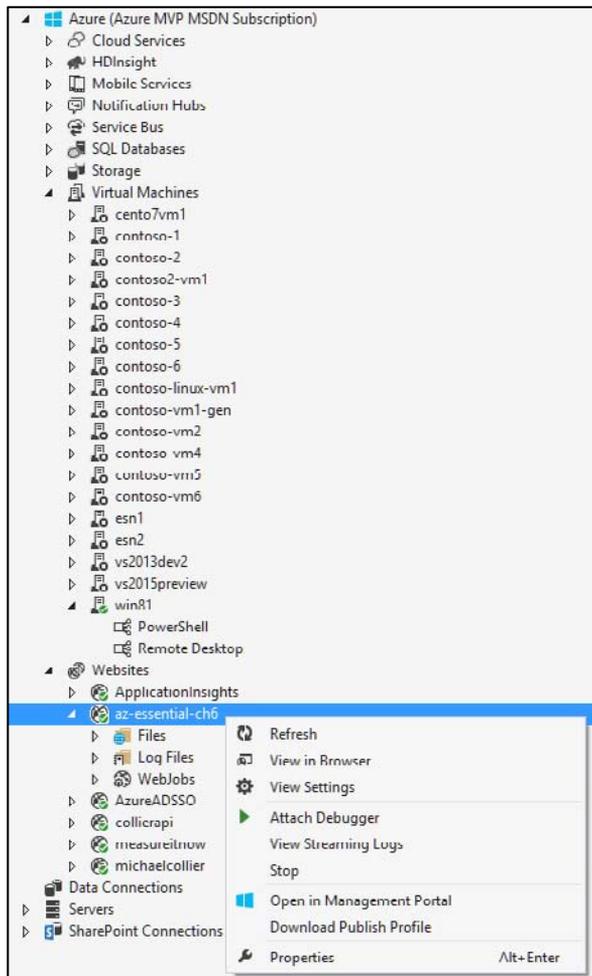


FIGURE 8-8 Details for the selected resource.

You can also create a new resource from Server Explorer. To do so, right-click the desired resource and select the option to create a new resource. For example, right-clicking the Websites node and selecting Create New Site opens the wizard to create a new Azure website, as seen in Figure 8-9.

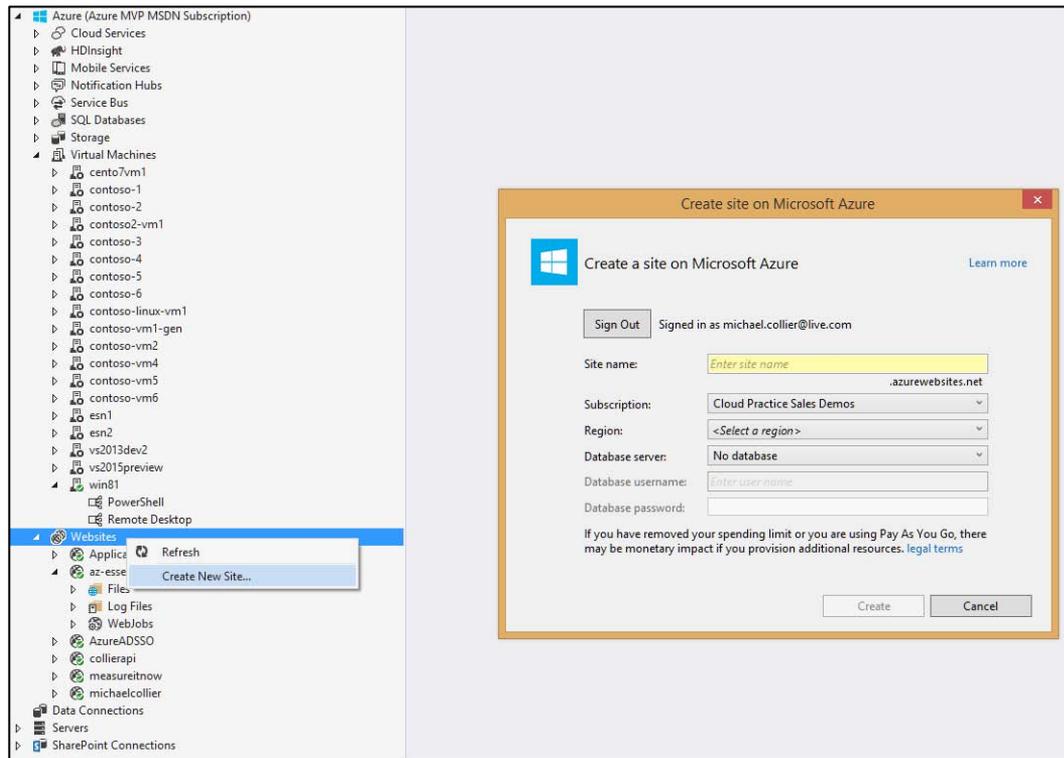


FIGURE 8-9 Create a new Azure website from Server Explorer.

Create an Azure resource

When the time comes to create a new Azure resource and write code to go with it, Visual Studio and the Azure SDK can be helpful tools. When creating a new Visual Studio project, navigate to the Cloud template section in the New Project wizard, as shown in Figure 8-10. From there, you can select the appropriate option, such as create a new Azure Cloud Service, create an Azure Mobile Service, and so on.

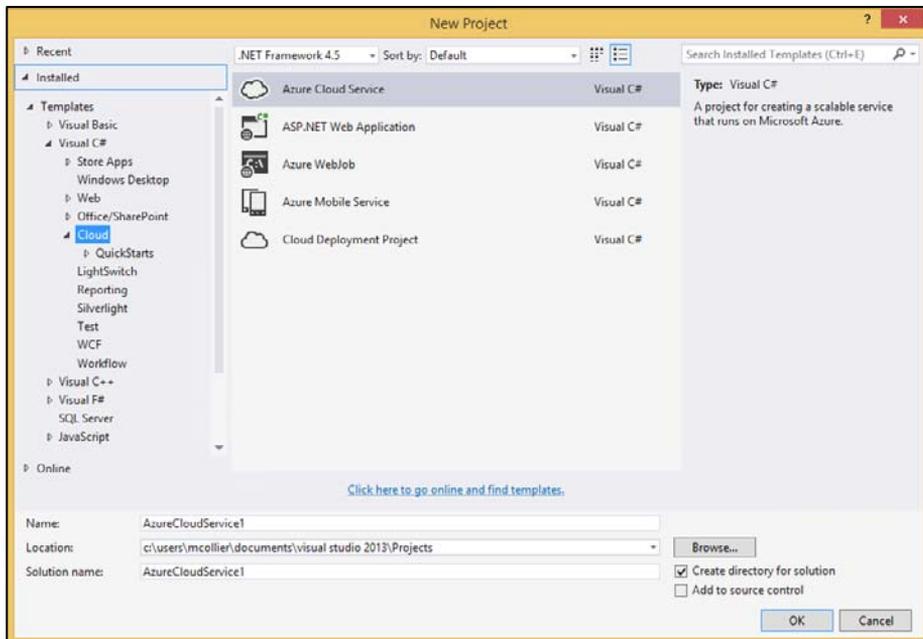


FIGURE 8-10 Create a new Cloud project.

The QuickStarts section, shown in Figure 8-11, provides a list of Microsoft-provided sample projects for specific Azure technologies. This can be a great way to learn basic features of a new or unfamiliar technology.

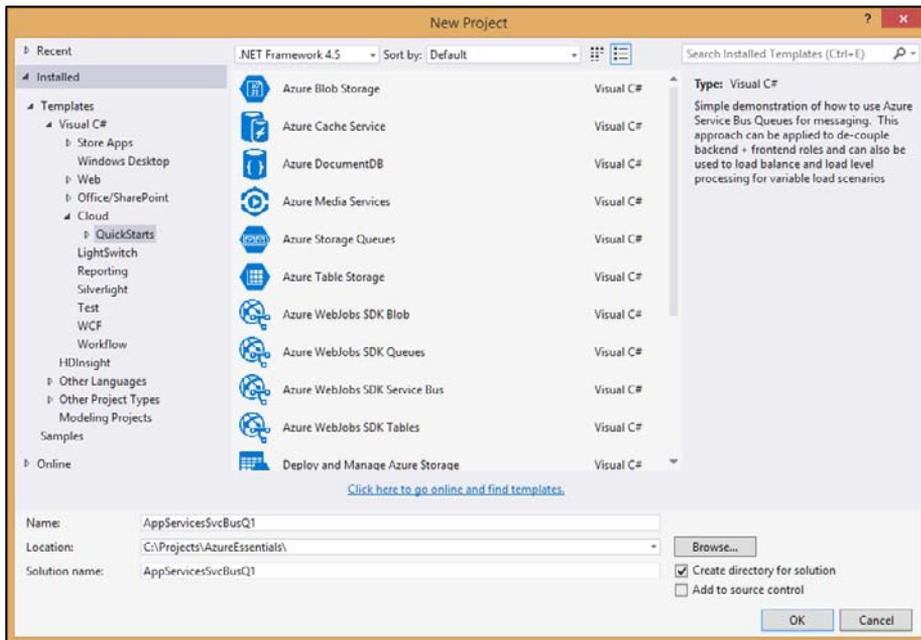


FIGURE 8-11 QuickStarts with Azure technologies.

Windows PowerShell

Visual Studio is an incredibly powerful tool for creating and managing solutions hosted in Azure. As great as it is, Visual Studio might not be the right tool for all scenarios. Instead, a scriptable tool such as PowerShell could be the right choice. Scenarios in which PowerShell might be preferred include the following:

- **Tool for IT professionals** Many IT professionals do not use Visual Studio for management of on-premises assets. It makes sense that they would not want to use Visual Studio for Azure-hosted assets, either. Instead, PowerShell is often one of the preferred tools, especially for managing Windows environments.
- **Automating provisioning and deployment of Azure resources** PowerShell provides a rich scripting environment for automating the provisioning and deployment of Azure resources. By using PowerShell scripts, you can automate the provisioning of new Azure virtual machines, create Azure Storage accounts, create Azure websites, create and import Azure Virtual Network settings, deploy Cloud Services, and much more.

Using PowerShell scripts to automate common tasks in Azure is also a great way to reduce potential errors. The scripts can be thoroughly tested, secured in a source control system, and repeatedly used with confidence that the same results will always be achieved.

If you have a task that you are going to perform repeatedly, it is beneficial to automate the task. It might take a greater time commitment initially to develop the script, but doing so will save a substantial amount of time every time you reuse it.

- **Accessing advanced or new Azure features not included in the Azure tools for Visual Studio or the management portal** The Azure PowerShell cmdlets include features that are not currently available in Visual Studio (for example, setting advance IP configuration for Azure Virtual Machines or nondefault configuration settings for the Azure Load Balancer). The update and release frequency for Azure PowerShell cmdlets is faster than that of the Azure SDK for Visual Studio. Because of this, you will often see new features appear in the PowerShell cmdlets (and REST API) before they appear in Visual Studio.

Additionally, new features are often released that are surfaced in the Azure PowerShell cmdlets but are not surfaced in either of the management portals (at least initially). This allows the Microsoft Azure product team to release the feature—and in some cases, fine-tune it—before releasing the feature to the management portals and any related UI elements. A prime example of this is the new Azure Files feature.

Azure PowerShell cmdlet installation

The Azure PowerShell cmdlets can be obtained one of two ways: either via the Web Platform Installer (Web PI) or by downloading the stand-alone installer from GitHub.

Earlier in this chapter, you saw the Azure Downloads page at <http://azure.microsoft.com/en-us/downloads/>, from which you were able to download the Azure SDK for .NET (Visual Studio). You can initiate Web PI from this same location, as seen in Figure 8-12, to install the Azure PowerShell cmdlets. Installing the Azure PowerShell cmdlets using Web PI could also install the latest Azure SDK, which might or might not be desired, depending on the Azure SDK version you are using for your current projects. If you want just the Azure PowerShell cmdlets, download the stand-alone installer from GitHub.

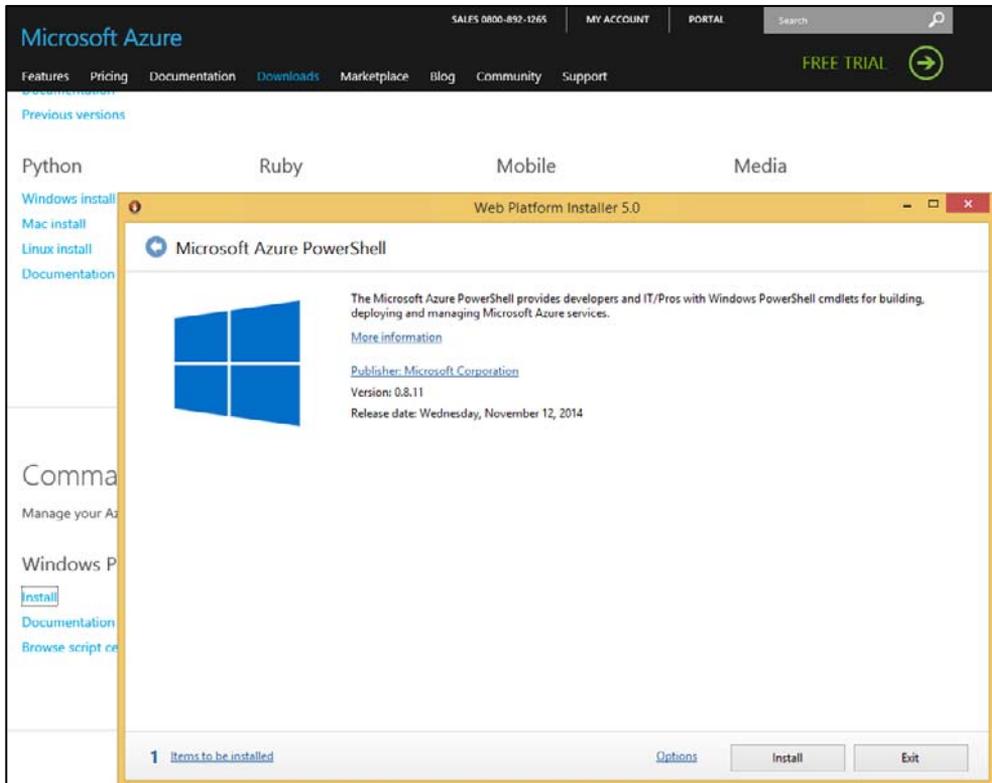


FIGURE 8-12 Azure PowerShell installation via Web Platform Installer.

Microsoft makes source code, issues, a wiki, and releases related to the Azure PowerShell cmdlets available on GitHub at <https://github.com/Azure/azure-powershell> (prior to December 2014, the repository was located at <https://github.com/Azure/azure-sdk-tools>). To obtain the stand-alone installer, go to the Releases section in the repository. There you will find a list of all releases, along with a link to download either the Windows Standalone installation package or the Web Platform Installer, as shown in Figure 8-13.

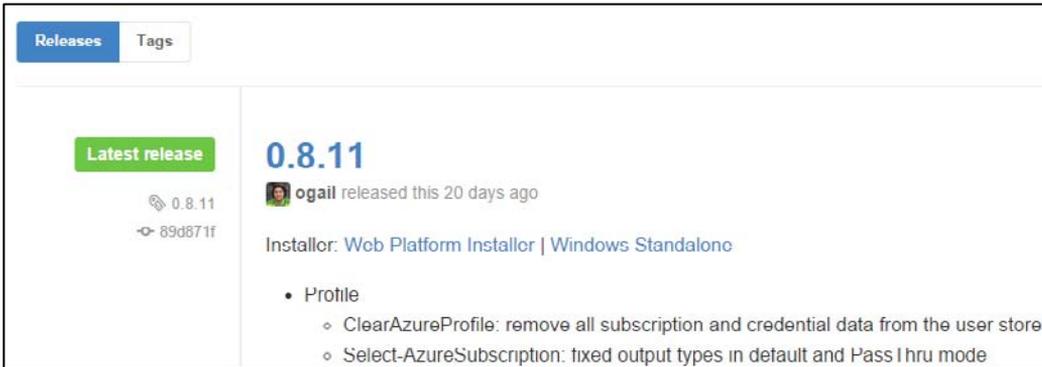


FIGURE 8-13 Azure PowerShell installation options from GitHub.

Clicking the Windows Standalone link in the GitHub repository will download a standard Windows MSI that can then be executed to install the Azure PowerShell cmdlets, as seen in Figure 8-14.



FIGURE 8-14 Windows stand-alone installation of Azure PowerShell.

Tip Be sure to read the detailed documentation on the Azure PowerShell GitHub page. The page contains a wealth of information on getting started with the Azure cmdlets, various features, and much more.

Connecting to Azure

To connect to Azure, you must have an Azure subscription. To sign up for a free trial (at <http://azure.microsoft.com>), you need a Microsoft account or a work or school (formerly organizational) account.

After installing the Azure PowerShell cmdlets, you will connect PowerShell to your Microsoft Azure subscription(s). There are two options for connecting, or authenticating, with Microsoft Azure: using a Microsoft account or using a management certificate.

Connecting using a Microsoft account

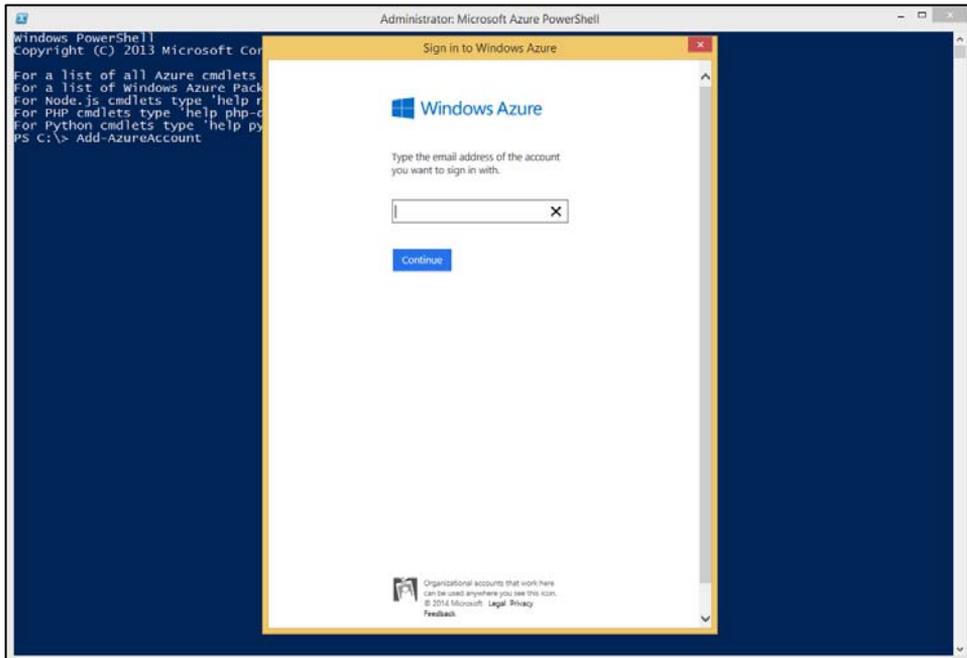
You can use your Microsoft account (for example, hotmail.com, outlook.com, and so on) or a work or school account (formerly organization account; for example, contoso.com) to access your Microsoft Azure subscription(s). For this method of connecting, Azure Active Directory (Azure AD) is used. This is the preferred method in many cases, because it is easier to manage access to a subscription (especially for a shared subscription that many people use). Additionally, using the Azure AD approach is required to work with the Azure Resource Manager API (see the section "Azure PowerShell modes" later in this chapter).

To connect to your Microsoft Azure subscription, follow these steps:

1. Open the Microsoft Azure PowerShell console. For a Windows 8 system, you can use the built-in Search to find PowerShell. Open the console by selecting the Microsoft Azure PowerShell option.



2. Enter the `Add-AzureAccount` command.
3. A dialog box prompts you for your email address and password. Your credentials are authenticated via Azure AD, and your subscription details are saved in your roaming user profile. An access token is also retrieved that allows PowerShell to access your Azure resources. This token is not permanent and will expire, at which point you will need to reauthenticate by using `Add-AzureAccount` again.



To view all available subscriptions, use the `Get-AzureSubscription` cmdlet. To change to a different subscription, use the `Select-AzureSubscription` cmdlet.

Tip Your profile data related to Azure PowerShell is stored in your user profile at `C:\Users\<user_name>\AppData\Roaming\Windows Azure Powershell`.

If you are using PowerShell for an automation script, you will want to avoid the pop-up window. In this case, use the `Credential` parameter to provide your credentials, as seen in the following example. The `Credential` parameter works for work or school accounts only.

```
$userName = "<your work or school account user name>"
$securePassword = ConvertTo-SecureString "<your work or school account password>"
-AsPlainText -Force

$cred = New-Object System.Management.Automation.PSCredential($userName, $securePassword)

Add-AzureAccount -Credential $cred
```

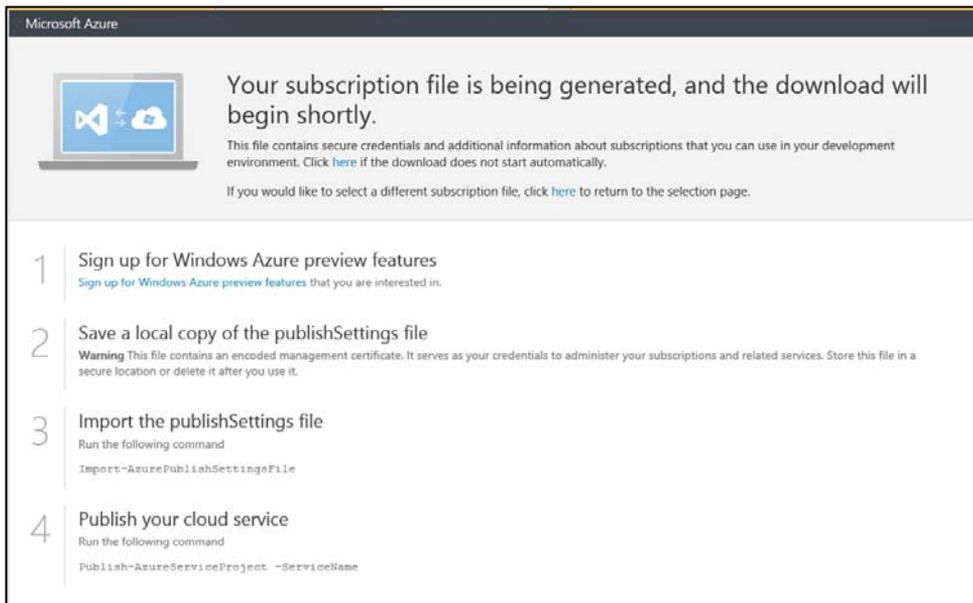
Connect using a management certificate

If you do not want to use the Azure AD approach previously mentioned, you can connect to your Azure subscription(s) using a management certificate. An Azure management certificate is an X.509 v3 certificate that authenticates a client application (for example, PowerShell, Visual Studio, custom code

you might write, and so on) that uses the Azure Service Management API.

The easiest way to obtain a management certificate is to let the Azure Management Portal generate one for you.

1. Open the Azure PowerShell console.
2. Enter the `Get-AzurePublishSettingsFile` command. This will open a webpage on the Azure Management Portal and will automatically download a `.publishsettings` file. You will need to log into the Azure Management Portal using your administrative credentials. The `.publishsettings` file will contain details about your Microsoft Azure subscription(s) and the necessary management certificate(s).



3. Save the `.publishsettings` file locally. Do not make this file public, because it contains the management certificate and subscription details. Anyone with access to this file can perform any action against the subscription(s).
4. Import the `.publishsettings` file using the `Import-AzurePublishSettingsFile`.

Tip The `.publishsettings` file contains sensitive information about your Microsoft Azure subscriptions, including the subscription ID and management certificate. Unauthorized users should not access the file. It is recommended that you delete the file after importing.

Alternatively, you can create a management certificate and set PowerShell to use it.

1. Open a Visual Studio command prompt as an administrator and execute the following command:

```
makecert -sky exchange -r -n "CN=<CertificateName>" -pe -a sha1 -len 2048 -ss My  
"<CertificateName>.cer"
```

Tip To find the Visual Studio command prompt, use Search in Windows 8 to find "Visual Studio tools." Open the shortcut folder to launch a Windows Explorer directory containing several shortcuts for Visual Studio tools, one of which is Developer Command Prompt for VS2013. Right-click the shortcut and select Run As Administrator.

For more information on the MAKECERT command, please refer to

[http://msdn.microsoft.com/en-us/library/windows/desktop/aa386968\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa386968(v=vs.85).aspx).

2. Upload the management certificate to Azure by using the Azure Management Portal. Go to the Settings page and then click MANAGEMENT CERTIFICATES, and then click UPLOAD on the bottom command bar.
3. Open the Microsoft Azure PowerShell console and execute the following commands:

```
$subscriptionName = '<SUBSCRIPTION_NAME>'  
$subscriptionId = '<SUBSCRIPTION_ID>'  
$thumbprint = '<MANAGEMENT_CERTIFICATE_THUMBPRINT>'  
$mgmtCert = Get-Item cert:\\CurrentUser\My\$thumbprint  
  
# Configure the subscription details in the Windows Azure PowerShell cmdlets  
Set-AzureSubscription -SubscriptionName $subscriptionName -SubscriptionId  
$subscriptionId -Certificate $mgmtCert
```

For more information on creating and using Azure management certificates, please refer to

<http://msdn.microsoft.com/en-us/library/azure/gg551722.aspx>.

Azure PowerShell modes

Azure contains two management APIs: the Azure Service Management API and the Azure Resource Manager API. The Azure Service Management API is the original management API. Many existing tools, such as Visual Studio, PowerShell, the Azure Management Portal, and other third-party tools, use the Azure Service Management API to programmatically interface with Azure. The Azure Resource Manager API is a new management API (at the time of this writing, it is still in Preview). Instead of working with individual resources (storage account, virtual machine, Azure SQL database, and so on), Azure Resource Manager allows you to group resources together by the solution you want to create, such as a blog, an e-commerce site, a data processing service, and so forth. The resources needed to

compose the solution are grouped together in a resource group.

For more information on Azure Resource Manager, please refer to the Microsoft Virtual Academy course at <http://www.microsoftvirtualacademy.com/training-courses/azure-resource-manager-devops-jump-start>.

The Azure PowerShell installation includes the ability to switch between these two management APIs. Two different PowerShell modules are installed: Azure, which works with the Service Management API, and AzureResourceManager, which works with the Azure Resource Manager API.

By default, the Azure module is imported into the current session when you open the Azure PowerShell console. To switch between modules, use the Switch-AzureMode cmdlet. For example, to switch to using Azure Resource Manager, execute the Switch-AzureMode AzureResourceManager cmdlet. To switch back to using the Azure Service Management API, execute the Switch-AzureMode AzureServiceManagement cmdlet.

Usage

There are many cmdlets available for working with the various Azure resources. To get a list of all the cmdlets, enter the command `get-help azure`.

Creating a new Azure virtual machine is a common task that can be simplified by using the Azure PowerShell cmdlets, as seen in the following example.

```
$DebugPreference = "Continue"

# Find the most recent Windows Server 2012 Datacenter image from the Azure image gallery
$image = Get-AzureVMImage `
  | where { $_.ImageFamily -ilike "Windows Server 2012 Datacenter" } `
  | sort PublishedDate -Descending `
  | select -ExpandProperty ImageName -First(1)

# Get the administrative credentials to use for the virtual machine
$credential = Get-Credential

# Create the new virtual machine, using the retrieved image and credentials
New-AzureQuickVM -Windows -ServiceName "azure-essential123" -Name "web-fe-1" `
  -ImageName $image -InstanceSize "Small" `
  -AdminUsername $credential.UserName -Password $credential.Password `
  -Location "East US"
```

Tip The preceding example sets the PowerShell variable `$DebugPreference = "Continue"`. Doing so can be useful when trying to diagnose why a command might not work as expected or might fail. With `$DebugPreference` set to "Continue", the underlying REST API request and response will be printed to the console.

Cross-Platform Command-Line Interface

For Windows users, the PowerShell cmdlets as discussed in the previous section are your best option for automating tasks and working from the command line, especially if you are scripting the provisioning of several Azure resources. However, for mixed environments, the Azure Cross-Platform Command-Line Interface (alternatively known as xplat-cli) provides a consistent experience for Linux, Mac OS, and Windows users alike.

Just like the Azure PowerShell project repository, the repository for the xplat-cli is available on GitHub, at <https://github.com/Azure/azure-xplat-cli>.

Installation

The xplat-cli is a Node.js application implemented using the Azure SDK for Node.js. Therefore, you will need to ensure Node.js is installed on your system.

Installing on Windows

If you do not have Node.js already installed on your system, you can use the Windows installer from the Microsoft Azure Downloads page (<http://azure.microsoft.com/en-us/downloads/>), as seen in Figure 8-15. Web PI will handle installing Node.js and the xplat-cli.

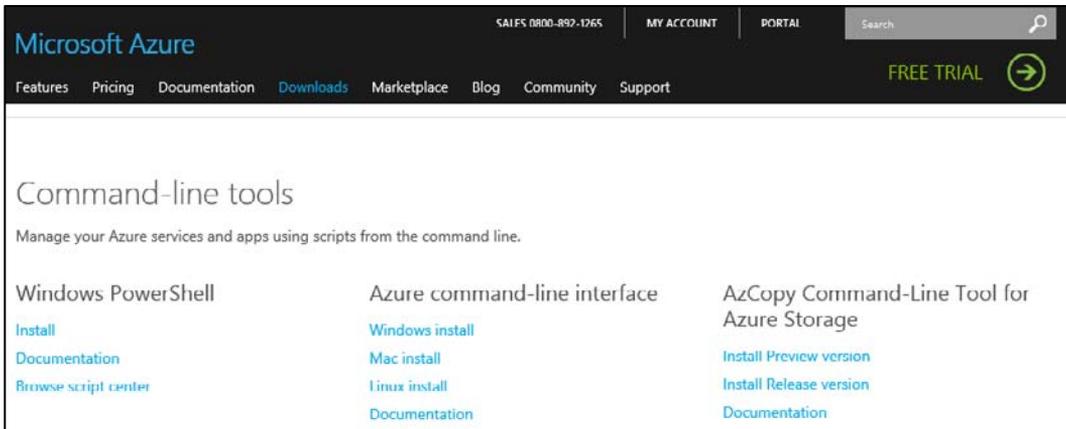


FIGURE 8-15 Install Azure command-line interface using Windows install.

Alternatively, you can install Node.js from the <http://nodejs.org> site. Just click INSTALL, as shown in Figure 8-16, to begin the process to install the latest version.

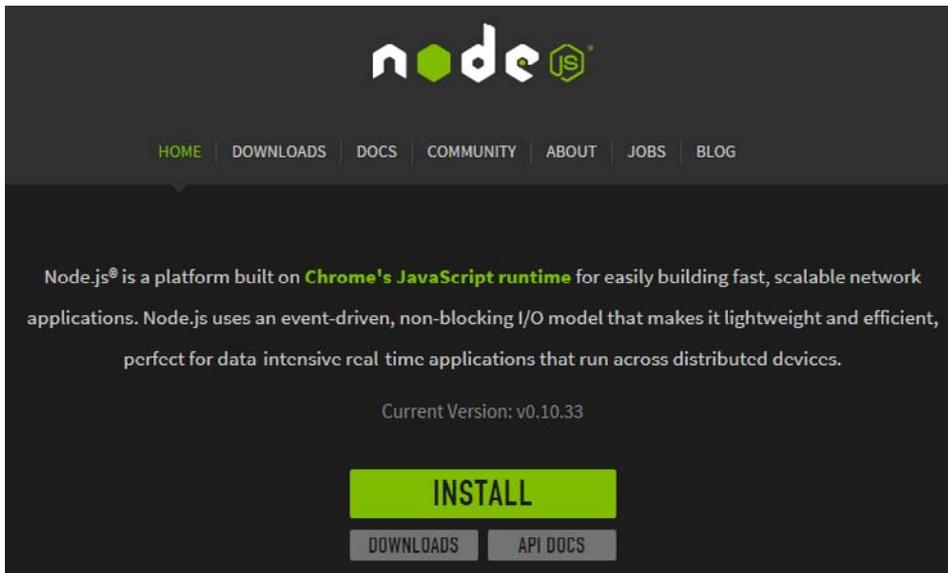
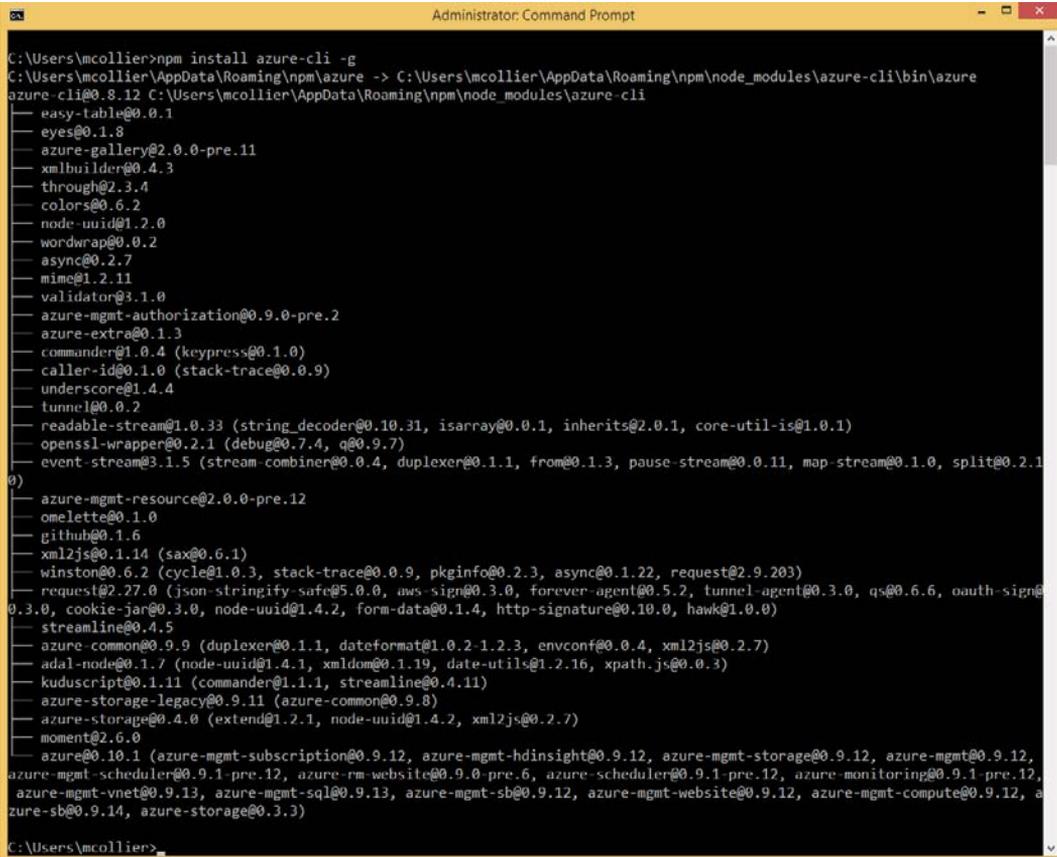


FIGURE 8-16 Install Node.js from the <http://nodejs.org> site.

Once Node.js is installed on your system, you can use npm (Node Package Manager) to install the xplat-cli.

```
npm install azure-cli -g
```

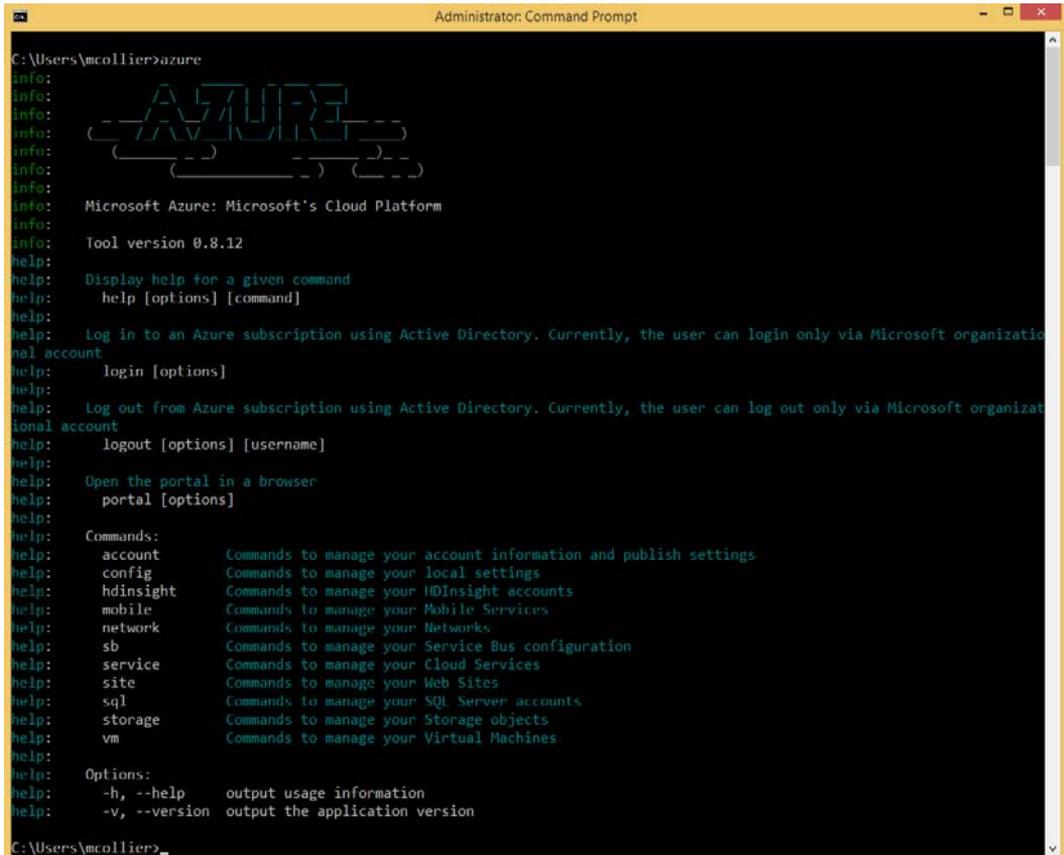
NPM will install all the necessary dependencies. When finished, you should see something similar to the results in Figure 8-17.



```
Administrator: Command Prompt
C:\Users\mcollier>npm install azure-cli -g
C:\Users\mcollier\AppData\Roaming\npm\azure -> C:\Users\mcollier\AppData\Roaming\npm\node_modules\azure-cli\bin\azure
azure-cli@0.8.12 C:\Users\mcollier\AppData\Roaming\npm\node_modules\azure-cli
├── easy-table@0.0.1
├── eyes@0.1.8
├── azure-gallery@2.0.0-pre.11
├── xmlbuilder@0.4.3
├── through@2.3.4
├── colors@0.6.2
├── node-uuid@1.2.0
├── wordwrap@0.0.2
├── async@0.2.7
├── mime@1.2.11
├── validator@3.1.0
├── azure-mgmt-authorization@0.9.0-pre.2
├── azure-extra@0.1.3
├── commander@1.0.4 (keypress@0.1.0)
├── caller-id@0.1.0 (stack-trace@0.0.9)
├── underscore@1.4.4
├── tunnel@0.0.2
├── readable-stream@1.0.33 (string_decoder@0.10.31, isarray@0.0.1, inherits@2.0.1, core-util-is@1.0.1)
├── openssl-wrapper@0.2.1 (debug@0.7.4, q@0.9.7)
├── event-stream@3.1.5 (stream-combiner@0.0.4, duplexlex@0.1.1, from@0.1.3, pause-stream@0.0.11, map-stream@0.1.0, split@0.2.1)
├── azure-mgmt-resource@2.0.0-pre.12
├── omelette@0.1.0
├── github@0.1.6
├── xml2js@0.1.14 (sax@0.6.1)
├── winston@0.6.2 (cycle@1.0.3, stack-trace@0.0.9, pkginfo@0.2.3, async@0.1.22, request@2.9.203)
├── request@2.27.0 (json-stringify-safe@5.0.0, aws-sign@0.3.0, forever-agent@0.5.2, tunnel-agent@0.3.0, qs@0.6.6, oauth-sign@0.3.0, cookie-jar@0.3.0, node-uuid@1.4.2, form-data@0.1.4, http-signature@0.10.0, hawk@1.0.0)
├── streamline@0.4.5
├── azure-common@0.9.9 (duplexlex@0.1.1, dateFormat@1.0.2-1.2.3, envconf@0.0.4, xml2js@0.2.7)
├── adal-node@0.1.7 (node-uuid@1.4.1, xmldom@0.1.19, date-util@1.2.16, xpath.js@0.0.3)
├── kuduscript@0.1.11 (commander@1.1.1, streamline@0.4.11)
├── azure-storage-legacy@0.9.11 (azure-common@0.9.8)
├── azure-storage@0.4.0 (extend@1.2.1, node-uuid@1.4.2, xml2js@0.2.7)
├── moment@2.6.0
├── azure@0.10.1 (azure-mgmt-subscription@0.9.12, azure-mgmt-hdinsight@0.9.12, azure-mgmt-storage@0.9.12, azure-mgmt@0.9.12, azure-mgmt-scheduler@0.9.1-pre.12, azure-rm-website@0.9.0-pre.6, azure-scheduler@0.9.1-pre.12, azure-monitoring@0.9.1-pre.12, azure-mgmt-vnet@0.9.13, azure-mgmt-sql@0.9.13, azure-mgmt-sb@0.9.12, azure-mgmt-website@0.9.12, azure-mgmt-compute@0.9.12, azure-sb@0.9.14, azure-storage@0.3.3)
C:\Users\mcollier>
```

FIGURE 8-17 Installing azure-cli.

When the installation is finished, you can launch the xplat-cli by entering `azure` at the command prompt. At this point, the xplat-cli will launch (although it might take a few seconds) and you will see "AZURE" in ASCII art, along with the version number and some basic help information, as shown in Figure 8-18.



```
C:\Users\mcollier>azure
info:
info:
info:
info:
info:
info:
info:
info: Microsoft Azure: Microsoft's Cloud Platform
info:
info: Tool version 0.8.12
help:
help: Display help for a given command
help: help [options] [command]
help:
help: Log in to an Azure subscription using Active Directory. Currently, the user can login only via Microsoft organization account
help: login [options]
help:
help: Log out from Azure subscription using Active Directory. Currently, the user can log out only via Microsoft organization account
help: logout [options] [username]
help:
help: Open the portal in a browser
help: portal [options]
help:
help: Commands:
help: account      Commands to manage your account information and publish settings
help: config       Commands to manage your local settings
help: hdinsight    Commands to manage your HDInsight accounts
help: mobile       Commands to manage your Mobile Services
help: network     Commands to manage your Networks
help: sb           Commands to manage your Service Bus configuration
help: service     Commands to manage your Cloud Services
help: site        Commands to manage your Web Sites
help: sql         Commands to manage your SQL Server accounts
help: storage     Commands to manage your Storage objects
help: vm          Commands to manage your Virtual Machines
help:
help: Options:
help: -h, --help  output usage information
help: -v, --version output the application version
C:\Users\mcollier>
```

FIGURE 8-18 Starting the xplat-cli on Windows.

Installing on Linux

Installing the xplat-cli on a Linux system is functionally similar to installing it on a Windows system. You will need to ensure you have Node.js installed. You can use the appropriate package manager for your Linux distribution (see <https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager>).

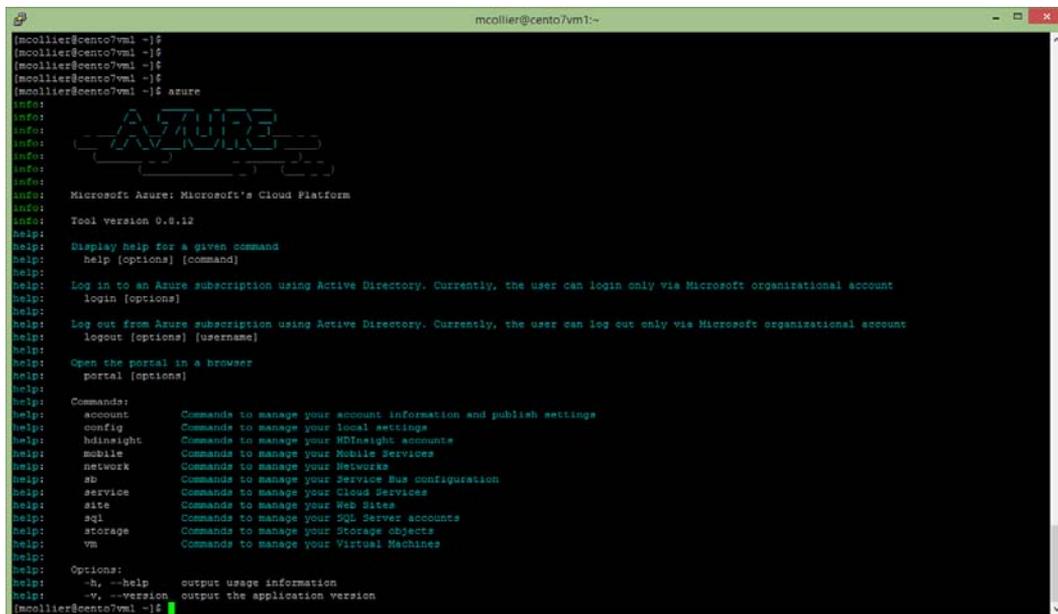
For example, on CentOS, execute the following command:

```
sudo curl -sL https://rpm.nodesource.com/setup | bash -
```

Then execute the following command:

```
sudo yum install -y nodejs
```

When finished, execute the azure command, as shown in Figure 8-19, which will yield the same results as executing the command on a Windows system.



```
mcollier@cento7vm1 ~
[mcollier@cento7vm1 ~]$
[mcollier@cento7vm1 ~]$
[mcollier@cento7vm1 ~]$
[mcollier@cento7vm1 ~]$
[mcollier@cento7vm1 ~]$ azure
info:
info:
info:
info:
info:
info: Microsoft Azure: Microsoft's Cloud Platform
info:
info: Tool version 0.9.12
help:
help: Display help for a given command
help: help [options] [command]
help:
help: Log in to an Azure subscription using Active Directory. Currently, the user can login only via Microsoft organizational account
help: login [options]
help:
help: Log out from Azure subscription using Active Directory. Currently, the user can log out only via Microsoft organizational account
help: logout [options] [username]
help:
help: Open the portal in a browser
help: portal [options]
help:
help: Commands:
help: account      Commands to manage your account information and publish settings
help: config       Commands to manage your local settings
help: hdinsight     Commands to manage your HDInsight accounts
help: mobile       Commands to manage your Mobile Services
help: network      Commands to manage your Networks
help: sb           Commands to manage your Service Bus configuration
help: service      Commands to manage your Cloud Services
help: site         Commands to manage your Web Sites
help: sql          Commands to manage your SQL Server accounts
help: storage      Commands to manage your Storage objects
help: vm           Commands to manage your Virtual Machines
help:
help: Options:
help: -h, --help    output usage information
help: -v, --version output the application version
[mcollier@cento7vm1 ~]$
```

FIGURE 8-19 Starting the xplat-cli on Linux (CentOS).

Connecting to Azure

Before you can begin to use the xplat-cli, you will need to connect to your Microsoft Azure subscription(s). The options for doing so with the xplat-cli are very similar to doing so with the Azure PowerShell cmdlets. There are two options: use a management certificate or use a work or school account.

Connect using a management certificate

Much like working with the Azure PowerShell cmdlets, the easiest way to get a management certificate is to let the Azure Management Portal create one. The management certificate comes in the form of a .publishsettings file.

1. If you do not already have a .publishsettings file, execute the following command to download one. This will open the Azure Management Portal and prompt you to sign in.

```
azure account download
```

2. Import the .publishsettings file by executing the following command (after changing to use the path to the file on your system):

```
Azure account import [path to your .publishsettings file]
```

Tip The .publishsettings file contains sensitive information about your Microsoft Azure subscriptions, including the subscription ID and management certificate. Unauthorized users should not access the file. It is recommended that you delete the file after importing.

Connect using a work or school account

Alternatively, you can use a work or school account (formerly organizational account) via Azure AD to connect to your Microsoft Azure subscription(s).

1. Execute the following command to log in using a work or school account:

```
azure login -u username -p password
```

2. Alternatively, execute the azure login command (without specifying the username or password) to log in interactively. This prevents credentials from being displayed on the screen in clear text.

Note This method will not work with a Microsoft account. If you are using a Microsoft account to manage your Azure subscription(s), you will need to create a new user in your default Azure AD directory and assign the user as a co-administrator for your Azure subscription. Adding administrative privileges for a user is covered in Chapter 1, “Getting started with Azure.”

Usage

Using the xplat-cli follows an intuitive “azure [topic] [verb] [options]” syntax; for example, azure account list or azure vm start web-fe-1.

Entering just the azure command will show you the top-level list of topics available (account, vm, service, and so on). To learn more about a specific topic, just enter azure followed by the topic. For example, azure vm will show the commands available for working with Azure Virtual Machines, as shown in Figure 8-20.



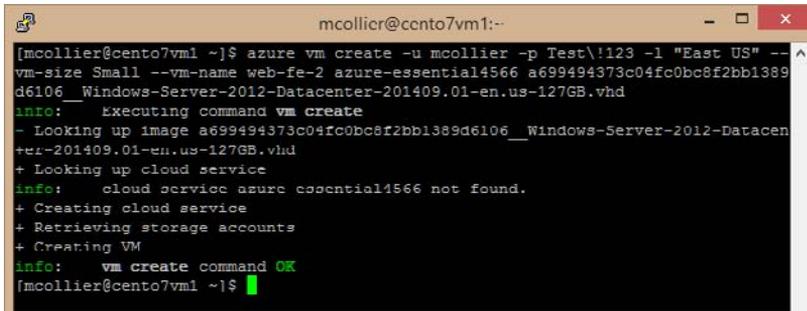
```
mcollier@cento7vm1:~]$ azure vm
help:  Commands to manage your Virtual Machines
help:
help:  Create a VM
help:    vm create [options] <dns-name> <image> [userName] [password]
help:
help:  Create a VM from json role file
help:    vm create-from [options] <dns-name> <role-file>
help:
help:  List the VM
help:    vm list [options]
help:
help:  Show details about the VM
help:    vm show [options] <name>
help:
help:  Delete the VM
help:    vm delete [options] <name>
help:
help:  Start the VM
help:    vm start [options] <name>
help:
help:  Restart the VM
help:    vm restart [options] <name>
help:
help:  Shutdown the VM
help:    vm shutdown [options] <name>
help:
help:  Capture the VM image
help:    vm capture [options] <vm-name> <target-image-name>
help:
help:  Export a VM to a file
help:    vm export [options] <vm-name> <file-path>
help:
help:  Commands to manage VM resource extensions
help:    vm extension list [options]
help:    vm extension set [options] <vm-name> <extension-name> <publisher-name>
help:    > <version>
help:    vm extension get [options] <vm-name>
help:
help:  Commands to manage your Docker Virtual Machine
help:    vm docker create [options] <dns-name> <image> <user-name> [password]
help:
help:  Commands to manage your Virtual Machine locations
help:    vm location list [options]
help:
```

FIGURE 8-20 Options for working with Azure Virtual Machines.

You can perform many tasks with the xplat-cli. For example, to create a new Azure virtual machine, you would execute the following command:

```
azure vm create -u mcollier -p Test!123 -l "East US" --vm-size Small
--vm-name web-fe-2 azure-essential4566
a699494373c04fc0bc8f2bb1389d6106__Windows-Server-2012-Datacenter-201409.01-en.us-127GB.vhd
```

When complete, you should see a result similar to that shown in Figure 8-21 that indicates the VM was created.



```
mcollier@cento7vm1:~$ az vm create -u mcollier -p Test\!123 -l "East US" --vm-size Small --vm-name web-fe-2 azure-essential4566 a699494373c04fc0bc8f2bb1389d6106_Windows-Server-2012-Datacenter-201409.01-en.us-127GB.vhd
info: Executing command vm create
- Looking up image a699494373c04fc0bc8f2bb1389d6106_Windows-Server-2012-Datacenter-201409.01-en.us-127GB.vhd
+ Looking up cloud service
info: cloud service azure essential4566 not found.
+ Creating cloud service
+ Retrieving storage accounts
+ Creating VM
info: vm create command OK
mcollier@cento7vm1 ~|$
```

FIGURE 8-21 The result of the `az vm create` command from the `xplat-cli`.

Chapter 9

Business Cases

There are many business cases for using Microsoft Azure: from spinning up temporary development and test environments to extending your on-premises infrastructure into the cloud or developing new applications that take advantage of the features available in Azure. In this chapter, we discuss a few common scenarios to give you some ideas for how you can use Azure.

Development and test scenarios

One of the common workloads in Azure is development and test (dev/test). In most cases, you can replicate all or part of your production infrastructure in Azure, whether it be on-premises or already running in Azure, and use the replica for development, staging, or testing.

If you have an on-premises datacenter and you want to set up a dev/test environment, you have to procure hardware, install the OS and the rest of the software, set up networking, configure the firewall, and so on. This can take a substantial amount of time. Once the testing is over, you have to either leave the hardware idle or repurpose it until you need it for other testing.

With Azure, you can provision what you need (VMs, cloud services, websites, storage, etc.) and proceed with the testing within minutes. When you are finished testing, you can tear down all of the services and stop paying for them. In fact, using Azure you can script the deployment and teardown of your dev/test environment.

Best of all, as your infrastructure grows, you can easily scale your dev/test environment to fit current needs. With an on-premises dev/test infrastructure, you have to go through the procurement and configuration process again.

If everything you have is on-premises, you can still use Azure for dev/test. You can set up a virtual network and extend your on-premises network into Azure. For example, you might want to test your application against a new version of SQL Server; you can have a web application running in your local datacenter that accesses SQL Server hosted in Azure.

If you have an MSDN subscription, you get a monthly credit to use for your dev/test infrastructure in Azure. In addition, several of the services are discounted. For example, VMs are discounted 33 percent. This can significantly lower the overall cost of setting up and using a dev/test infrastructure.

Here are some other business cases that you can cover using Azure to quickly replicate the parts of your infrastructure.

- **You have the flexibility of trying something small really fast** Let's say you only want to test one thing. For example, you want to change the way something is displayed in your website, but you're not sure if it will work or how it will work. You can make the modifications to the website and then deploy it as a new website with the configuration pointing to the production backend. Then you can check the workflow and visual layout and decide if it is worth setting up a complete dev/test environment to proceed.
- **Load testing** You can create an entire copy of your production environment and then do load testing on the copy. This can include different cloud services, VMs, websites, storage, virtual networks, and so on. This gives you the isolation to do load testing without affecting any of the production services, and it can help you pinpoint potential bottlenecks in your workflow so you can handle them before they affect the customer.

You can use load testing to figure out the scope of the resources you need to handle different loads, such as the size of the VMs or the number of cloud service instances. You can then change the compute services to do autoscaling where possible. For example, you might discover that as the percentage of CPU utilization exceeds 60 percent, your website becomes unacceptably slow, so you decide to implement autoscaling to increase the instance count when the CPU utilization hits the target value.

Load testing leads to a better overall experience for your customer.

- **Software upgrades** If you are using software from an external company, you will want to test your software with the company's software for compatibility before upgrading your production services. For example, if you use SQL Server 2012, and SQL Server 2014 is released, you will want to test your application against the new version before upgrading. You might need to modify your software to work with the new version of SQL Server and go through the whole cycle of testing, staging, and implementation.

In an on-premises environment, you can probably get a prerelease copy or a free short-term trial of the new version. However, you need to have infrastructure on which to run it, so as with the previous examples, you might need to procure hardware and so on. With Azure, the software might be available in a preconfigured VM, as is the case with Windows, SQL Server, Oracle, and Linux, among others. In that case, you can just provision a new VM with the new version in a dev/test scenario and run your software against it.

If there is no preconfigured VM in Azure, you can provision a Windows or Linux VM, install the new version of software on it, and use that for your dev/test scenario.

- **A/B testing** Let's say you want to perform some A/B testing on your website without repeatedly redeploying the different versions of the website. Azure Websites allows multiple deployment slots. You can publish version A to one slot and version B to another and then swap

them in and out of production as needed to perform the testing and metrics collection.

Another option is to use the weighted round-robin load balancing available with Azure Traffic Manager. Traffic Manager gives you the ability to balance incoming traffic among multiple services using the following load balancing methods: performance, round robin, and failover. With round-robin load balancing, you can weight the different deployments. This means you can divert a small percentage of traffic to a separate deployment to perform A/B testing.

For more information about using Traffic Manager load balancing, check out <http://msdn.microsoft.com/en-us/library/azure/dn339010.aspx>. For information specifically about weighted round-robin load balancing, please refer to this article: <http://azure.microsoft.com/blog/2014/06/26/azure-traffic-manager-external-endpoints-and-weighted-round-robin-via-powershell/>.

Hybrid scenarios

The number of companies running solutions in the cloud is increasing at an incredible rate. Their success encourages other organizations to take the same step. Some organizations will not be able to move all of their workloads into the cloud, either because of regulatory issues or because some workloads cannot run in a virtualized environment. In these cases, hybrid computing, in which a company runs part of its infrastructure in the cloud and part on premises, will be an important strategy.

The Microsoft Azure platform provides a great hybrid computing story. There are multiple ways to connect an on-premises datacenter to one or more Azure regions. As discussed in Chapter 5, “Azure Virtual Networks,” Azure provides both site-to-site and point-to-site virtual network connectivity. Either option provides a secure VPN connection between on-premises assets and resources hosted in Azure. An additional hybrid connectivity option is Azure ExpressRoute, which enables a private connection between Azure and your on-premises infrastructure or colocation facility, all without going over the public Internet.

Network connectivity

Irrespective of the chosen option—site-to-site, point-to-site, or ExpressRoute—hybrid connectivity is a key scenario for the Azure platform. Creating a hybrid connection opens a wide range of possibilities to extend an on-premises infrastructure to the cloud. Two common scenarios for network-enabled hybrid connectivity are the following:

- **Hosting a website in Azure, but keeping the database on premises** In an organization’s journey to the cloud, migrating the on-premises data to Azure can be one of the more difficult tasks. The difficulty usually comes in one of two forms: a technical issue or a compliance requirement. On the technical front, as an example, the application in question is designed to use a database that is not supported in Azure. On the compliance front, perhaps there is a regulatory requirement that cannot be met with Azure SQL Database or by running a database

(SQL Server, MongoDB, and so on) on Azure Virtual Machines. In these cases, an organization might choose to host the website in Azure using Azure Websites or Azure Cloud Services (via a web role) with the database remaining on premises. Connectivity between the website and the database could then be established using one of the aforementioned technologies: a site-to-site connection, a point-to-site connection, or ExpressRoute.

- **Accessing an on-premises service** Sometimes a website has a dependency on a particular service that cannot be moved to the cloud. Perhaps the website depends on an API that performs a crucial business calculation, and that API cannot be moved because of security, because other on-premises services also depend on the service, or because it is legacy technology that is not supported in Azure. In such a scenario, a hybrid connection is established between Azure and the on-premises infrastructure to allow the Azure-hosted website to freely communicate with the necessary API that continues to reside on premises.

Besides using a network connection in this scenario, an Azure Service Bus Relay could be used to access an on-premises service. For information on how to use the Azure Service Bus Relay service, please refer to

<http://azure.microsoft.com/en-us/documentation/articles/service-bus-dotnet-how-to-use-relay/>.

Internet connectivity

There are many scenarios in which all that is needed is an Internet connection rather than a special hybrid connectivity solution. After all, the ability to connect to Internet-accessible services is one of the attractive features of cloud computing. A few common scenarios include these:

- **Storage of archival data** Large amounts of data, especially archival data that is rarely accessed, can be very expensive to store on premises. The cost in terms of infrastructure, people, software licenses, and physical space can quickly put a tremendous financial burden on an organization. As discussed in Chapter 4, “Azure Storage,” Azure provides virtually limitless storage capacity at an incredibly low price. An organization might wish to use the scalable storage provided by Azure Blob storage as a data archival store. When the data is needed, the on-premises service(s) download the data from Azure Blob storage and perform the necessary processing. A basic Internet connection will often suffice, but an ExpressRoute connection could also be used for improved speed and security.

Another option for storage of archival data is Microsoft Azure StorSimple. StorSimple includes a hardware appliance that is installed on premises. The appliance keeps frequently accessed data local (on the device). As data ages (is less frequently accessed), it is automatically moved to Azure Blob storage. For more information on StorSimple, please refer to <http://azure.microsoft.com/en-us/documentation/services/storsimple/>.

- **Azure Active Directory** As discussed in Chapter 7, “Azure Active Directory,” organizations can choose to synchronize their Azure AD users and groups with user and group information

from their on-premises Active Directory. In doing so, they can choose to use Azure Active Directory Sync (AADSync; previously known as DirSync) to synchronize the user data and a password hash, making Azure AD the authority for user authentication. Alternatively, an organization might wish to synchronize the user data but require users to authenticate via an Active Directory Federation Services (AD FS) endpoint residing on premises, effectively redirecting the user to an on-premises AD FS site for authentication before redirecting to the desired location.

- **Burst to the cloud** Sometimes an organization's on-premises infrastructure is not able to handle the required load. Maybe there is a holiday season rush or a government-mandated period to sign up for an important service. Instead of building the on-premises infrastructure to handle the temporary surge in demand, an organization might choose to leverage the elastic nature of the cloud to burst to the cloud when needed and scale back to only on-premises services when the load returns to normal. In this scenario, an organization could use Azure Websites, Cloud Services, or Azure Virtual Machines to host the service and could implement autoscale rules to ensure capacity keeps up with user demand.

Application and infrastructure modernization and migration

There comes a time in every application's life when it is time to upgrade. It could be a user interface redesign or a hardware refresh. The Azure platform cannot help create an appealing, modern user interface, but it can modernize the supporting infrastructure.

Many organizations will go through a periodic hardware refresh cycle; typically, this happens about every three years. When it is time for a hardware refresh, organizations today have a new question to ask: Should we buy new on-premises hardware, or should we leverage our infrastructure and services to the cloud?

Besides a required hardware refresh, an organization might choose to migrate to the cloud because it has reached physical capacity limits in its existing on-premises datacenter or because it is going to in the very near future. Perhaps the current datacenter does not have enough physical space for more servers or cannot supply the necessary power or cooling. Maybe there is a desire to eliminate or reduce the management of hardware infrastructure going forward. Moving to the cloud might enable the organization to get out of the datacenter business completely, or at least partially (see the section "Hybrid scenarios" earlier in this chapter). In this case, Microsoft is responsible for the hardware and related infrastructure components of the datacenter, and the organization can focus on providing great business solutions.

Some organizations will choose to migrate to the cloud to get capacity in new geographies they can't currently support because they have no presence in that area or because it would be cost-prohibitive. There are Azure datacenters in 19 regions around the world (including two regions in China operated and sold by 21Vianet). Instead of building and maintaining a global datacenter presence, an organization can elect to take advantage of Microsoft's existing investments and deploy

to multiple regions with ease.

See Also For the current list of Azure regions, please refer to <http://azure.microsoft.com/en-us/regions/>.

Should the choice be to modernize or migrate to the cloud, there is certainly a wealth of Azure resources available. In choosing to adopt these resources, an organization could have many questions to answer, including these:

- Do we leverage PaaS, IaaS, or both?
- Instead of maintaining a custom solution, should we leverage platform-provided services such as Azure Search or Azure Media Services?
- Should we move everything, or just some components? What hybrid model works best for our requirements?
- Which Azure region(s) should be used?
- How does using Azure affect our business and operations model?
- What is our SLA? What is our disaster recovery story?

Azure Mobile Services

In today's world, mobile devices—from tablets to phones to watches to fitness bands—are everywhere you look. Having a mobile application can be a big plus for a company, whether it's used externally, internally, or both.

Microsoft Azure Mobile Services is a Backend-as-a-Service that provides multiple features to make it easier and quicker to create a mobile application. Mobile Services is both flexible and scalable, so when your application becomes widely used, you can scale appropriately to handle your customers' needs.

Another advantage of Azure Mobile Services is that you only have to write one version of your backend. The backend can be used by devices running iOS, Android, and Windows, allowing you to reach every user on every platform without extra work.

The following are some of the features provided by Azure Mobile Services. You can certainly program a service to implement these features from the ground up, but using Azure Mobile Services saves you the time and money it would take to do that.

- **Data storage** You can choose your data storage to be powered by SQL Database, which has an interface simple enough to use without being a DBA. You can also integrate with SQL Server, Oracle, SAP, MongoDB, and Azure Table storage.

You can write your application to work offline and synchronize the data when the application can go online again. This is helpful when the customer loses Internet connectivity—he or she can continue to work, knowing the work will be stored on the backend when connectivity is

regained.

- **User authentication and data authorization are greatly simplified** You can easily implement SSO with Azure AD, Microsoft, Facebook, Twitter, and Google.
- **Push notifications** You can send information for customer and enterprise applications to any customer's mobile device by using Microsoft Azure Notification Hubs. This can come from any backend, whether it runs in Azure or is on premises. Notification Hubs automatically handles the server-side code to push messages to the push notification services for iOS, Android, and Windows Phone devices.

Notification Hubs has a tagging feature that can be used to target audiences based on activity, interest, location, or preference. In addition, the templates feature of Notification Hubs enables you to send localized push notifications in the customer's own language.

- **Because Mobile Services runs in Azure, you can easily scale in and out to meet customer demand** You can even set up autoscaling that will automatically scale out as demand increases, handling millions of devices.
- **You can use Microsoft Azure Scheduler to perform backend processing on the server at a scheduled time** For example, you might want to create a scheduled job that requests an update from your on-premises database and stores the new information in a table, waiting to be retrieved by your mobile application.
- **You can create a hybrid connection** This connection can be used to connect the mobile application to on-premises systems, Office 365, and SharePoint.

About the authors



Robin E. Shahan is a Microsoft Azure MVP with over 25 years of experience developing complex, business-critical applications for Fortune 100 companies. As the VP of Technology for the startup GoldMail (DBA PointAcross), she re-architected their entire infrastructure and migrated it to Microsoft Azure, reducing their costs by 90%. Robin is the President of Nightbird Consulting, focusing on helping companies architect and develop scalable and efficient solutions utilizing the Azure platform.

Robin regularly speaks about Microsoft Azure at various .NET User Groups and Code Camps and runs the San Francisco Azure meetup. She can be found on Twitter as [@RobinDotNet](https://twitter.com/RobinDotNet), and you can read her articles about Microsoft Azure (and other subjects) at <http://robindotnet.wordpress.com>.



Michael S. Collier is a five-time Microsoft Azure MVP and most recently served as a Principal Cloud Architect with Aditi Technologies. He has over 13 years of experience with various consulting and technology firms where he was instrumental in leading and developing solutions for a wide range of clients. He has a vast amount of experience in helping companies determine the best strategy for adopting cloud computing, and providing the insight and hands-on experience to ensure they are successful. Michael is also a respected technology community leader and is often found sharing his Microsoft Azure insights and experiences at regional and national conferences. You can follow Michael's

experiences with Azure on his blog at <http://www.michaelscollier.com> and on Twitter at [@MichaelCollier](https://twitter.com/MichaelCollier).

Michael lives in Marysville, Ohio. He is a 2003 graduate of The Ohio State University and is a passionate Buckeyes fan. Michael is also an avid golfer, although golf doesn't always like him.



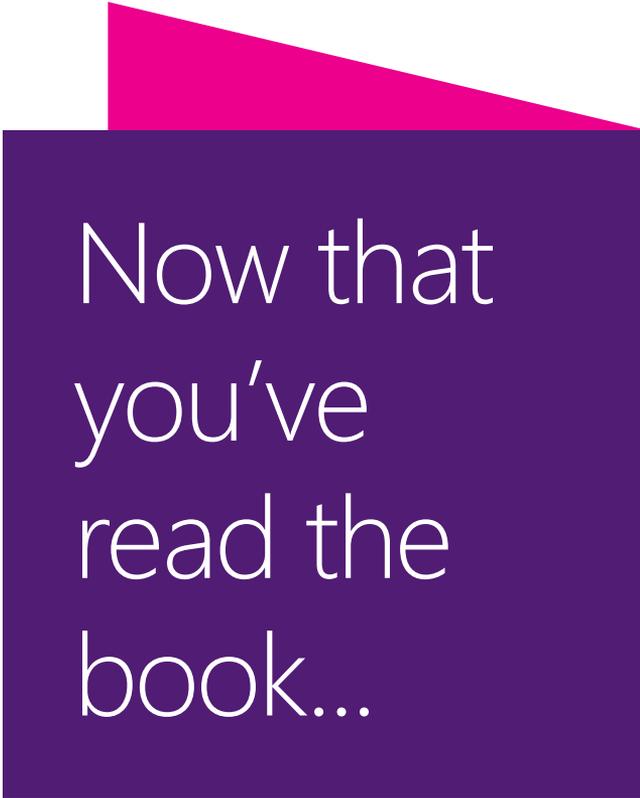
From technical overviews to drilldowns on special topics, get *free* ebooks from Microsoft Press at:

www.microsoftvirtualacademy.com/ebooks

Download your free ebooks in PDF, EPUB, and/or Mobi for Kindle formats.

Look for other great resources at Microsoft Virtual Academy, where you can learn new skills and help advance your career with free Microsoft training delivered by experts.

Microsoft Press



Now that
you've
read the
book...

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!

