Microsoft

OFFICIAL MICROSOFT LEARNING PRODUCT

20532D

Developing Microsoft Azure Solutions

Product Number: 20532D

Part Number: X21-64488

Released: 04/2018

MICROSOFT LICENSE TERMS
MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which includes the media on which you received it, if any. These license terms also apply to Trainer Content and any updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms apply.

BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below for each license you acquire.

1. DEFINITIONS.

a. "Authorized Learning Center" means a Microsoft IT Academy Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.

b. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.

c. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns or controls that is located at an Authorized Learning Center's training facilities that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.

d. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.

e. "Licensed Content" means the content accompanying this agreement which may include the Microsoft Instructor-Led Courseware or Trainer Content.

f. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program.

g. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals and developers on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics or Microsoft Business Group courseware.

h. "Microsoft IT Academy Program Member" means an active member of the Microsoft IT Academy Program.

i. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.

j. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals and developers on Microsoft technologies.

k. "MPN Member" means an active Microsoft Partner Network program member in good standing.

l. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.

m. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.

n. "Trainer" means (i) an academically accredited educator engaged by a Microsoft IT Academy Program Member to teach an Authorized Training Session, and/or (ii) a MCT.

o. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.

2. USE RIGHTS. The Licensed Content is licensed not sold. The Licensed Content is licensed on a ***one copy per user basis***, such that you must acquire a license for each individual that accesses or uses the Licensed Content.

2.1 Below are five separate sets of use rights. Only one set of rights apply to you.

a. **If you are a Microsoft IT Academy Program Member:**
   i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
   ii. For each license you acquire on behalf of an End User or Trainer, you may either:
      1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
      2. provide one (1) End User with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
      3. provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content,

   provided you comply with the following:
   iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
   iv. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
   v. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
   vi. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,

vii. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Authorized Training Sessions,
viii. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and
ix. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

b. **If you are a Microsoft Learning Competency Member**:
i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
ii. For each license you acquire on behalf of an End User or Trainer, you may either:
    1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Authorized Training Session and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**
    2. provide one (1) End User attending the Authorized Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
    3. you will provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content,

    provided you comply with the following:
iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
iv. you will ensure that each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
v. you will ensure that each End User provided with a hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
vi. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
vii. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for your Authorized Training Sessions,
viii. you will only use qualified MCTs who also hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Authorized Training Sessions using MOC,
ix. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
x. you will only provide access to the Trainer Content to Trainers.

c. **If you are a MPN Member**:

i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

ii. For each license you acquire on behalf of an End User or Trainer, you may either:

1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
2. provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
3. you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content,

provided you comply with the following:

iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,

iv. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,

v. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

vi. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,

vii. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,

viii. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,

ix. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and

x. you will only provide access to the Trainer Content to Trainers.

d. **If you are an End User:**

For each license you acquire, you may use the Microsoft Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led Courseware is in digital format, you may access the Microsoft Instructor-Led Courseware online using the unique redemption code provided to you by the training provider and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

e. **If you are a Trainer.**

i. For each license you acquire, you may install and use one (1) copy of the Trainer Content in the form provided to you on one (1) Personal Device solely to prepare and deliver an Authorized Training Session or Private Training Session, and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Trainer Content. You may not install or use a copy of the Trainer Content on a device you do not own or control. You may also print one (1) copy of the Trainer Content solely to prepare for and deliver an Authorized Training Session or Private Training Session.

ii. You may customize the written portions of the Trainer Content that are logically associated with instruction of a training session in accordance with the most recent version of the MCT agreement. If you elect to exercise the foregoing rights, you agree to comply with the following: (i) customizations may only be used for teaching Authorized Training Sessions and Private Training Sessions, and (ii) all customizations will comply with this agreement. For clarity, any use of "*customize*" refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

2.2 **Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.

2.3 **Redistribution of Licensed Content**. Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.

2.4 **Third Party Notices**. The Licensed Content may include third party code tent that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code ntent are included for your information only.

2.5 **Additional Terms**. Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.

3. LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY. If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("**Pre-release**"), then in addition to the other provisions in this agreement, these terms also apply:

a. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.

b. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its technology, technologies, or products to third parties because we include your feedback in them. These rights survive this agreement.

c. **Pre-release Term**. If you are an Microsoft IT Academy Program Member, Microsoft Learning Competency Member, MPN Member or Trainer, you will cease using all copies of the Licensed Content on the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using the Licensed Content on the Pre-release technology, or (ii) sixty (60) days after the commercial release of the technology that is the subject of the Licensed Content, whichever is earliest ("**Pre-release term**"). Upon expiration or termination of the Pre-release term, you will irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.

4. **SCOPE OF LICENSE**. The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
   - access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
   - alter, remove or obscure any copyright or other protective notices (including watermarks), branding or identifications contained in the Licensed Content,
   - modify or create a derivative work of any Licensed Content,
   - publicly display, or make the Licensed Content available for others to access or use,
   - copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
   - work around any technical limitations in the Licensed Content, or
   - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation.

5. **RESERVATION OF RIGHTS AND OWNERSHIP**. Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.

6. **EXPORT RESTRICTIONS**. The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.

7. **SUPPORT SERVICES**. Because the Licensed Content is "as is", we may not provide support services for it.

8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.

9. **LINKS TO THIRD PARTY SITES**. You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.

10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.

11. **APPLICABLE LAW.**
    a. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

b. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.

12. LEGAL EFFECT. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

13. DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

14. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US$5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 $ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.
Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et.
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised July 2013

Welcome!

Thank you for taking our training! We've worked together with our Microsoft Certified Partners for Learning Solutions and our Microsoft IT Academies to bring you a world-class learning experience—whether you're a professional looking to advance your skills or a student preparing for a career in IT.

- **Microsoft Certified Trainers and Instructors**—Your instructor is a technical and instructional expert who meets ongoing certification requirements. And, if instructors are delivering training at one of our Certified Partners for Learning Solutions, they are also evaluated throughout the year by students and by Microsoft.

- **Certification Exam Benefits**—After training, consider taking a Microsoft Certification exam. Microsoft Certifications validate your skills on Microsoft technologies and can help differentiate you when finding a job or boosting your career. In fact, independent research by IDC concluded that 75% of managers believe certifications are important to team performance[1]. Ask your instructor about Microsoft Certification exam promotions and discounts that may be available to you.

- **Customer Satisfaction Guarantee**—Our Certified Partners for Learning Solutions offer a satisfaction guarantee and we hold them accountable for it. At the end of class, please complete an evaluation of today's experience. We value your feedback!

We wish you a great learning experience and ongoing success in your career!

Sincerely,

Microsoft Learning
www.microsoft.com/learning

Microsoft | Learning

[1] *IDC*, Value of Certification: Team Certification and Organizational Performance, *November 2006*

Acknowledgments

Microsoft Learning would like to acknowledge and thank the following for their contribution towards developing this title. Their effort at various stages in the development has ensured that you have a good classroom experience.

Sidney Andrews – Subject Matter Expert/Content Developer

Sidney Andrews is an Azure MVP, a Microsoft Certified Training Regional Lead and owner of SeeSharpRun.NET. Sidney has authored multiple developer and Azure courses available on http://edx.org and has also been featured as a presenter on Channel 9, Ignite and Build. His specialties include Azure, XAML, C#, and TypeScript.

Tiago Costa – Content Developer

Tiago Costa is an IT Consultant in Microsoft Technologies. For the past 15 years, he has been architecting and developing solutions for clients using Azure, Office 365, SharePoint, .Net and SQL Server. He has robust real-world experience and regularly teaches Azure, Office 365, SharePoint, .Net, and SQL Server classes across Europe and other worldwide locations.

Tiago has retained several Microsoft Certifications and is also a Microsoft Certified Trainer. In 2013 he was nominated MCT Regional Lead by Microsoft Corp and had been renewed every year. In 2016 he has been awarded the MVP Award in Office Server and Services for his community efforts in sharing his knowledge and experience. This award gives him the opportunity to go even further in getting more opportunities to share passion for Cloud, SharePoint, and Office 365 Development.

Courtney Andrews – Content Reviewer

Courtney Andrews is a content reviewer and maintainer for various http://edx.org Azure courses released by SeeSharpRun.NET. He has edited and reviewed courses ranging from Azure architecture to Angular, HTML5, and the Microsoft data platform. Courtney is also a regular contributor to the GitHub repositories for 20532 and 20535.

Contents

About This Course

This section provides a brief description of the course, audience, suggested prerequisites, and course objectives.

Course Description

This course offers students the opportunity to take an existing web application and expand its functionality as part of moving it to Azure. The course does not require any existing experience with the ASP.NET platform but does require experience with C#. This course focuses on the development considerations and decisions necessary when building a highly available solution in the cloud. This course also prepares the students for the 70-532: Developing Microsoft Azure Solutions certification exam.

Audience

This course is intended for students who have experience building web applications. Students should also have experience with the Microsoft Azure platform and a basic understanding of the services offered in Azure.

The candidates targeted by this training have basic experience in implementing and monitoring Microsoft Azure solutions. Candidates are also proficient with the development tools, techniques and approaches used to build application solutions using C#.

Student Prerequisites

This course requires that students have the following knowledge and skills:

- Compare the services available in the Azure platform.
- Configure and deploy web applications.
- Create Azure Web Apps.
- Create and configure Virtual Machines.
- Create a Virtual Network.
- Create a Storage account.
- Manage blobs and containers in a Storage account.
- Create, configure and connect to an Azure SQL Database instance.
- Manage users, groups and subscriptions in an Azure Active Directory instance.

Course prerequisites can be met by having knowledge equivalent to, or by attendance at, courses 10979D: *Microsoft Azure Fundamentals* and 20483D: *Programming in C#* as this current course will build upon knowledge and skills covered in both courses.

Course Objectives

After completing this course, students will be able to:

- Compare the services available in the Azure platform.
- Configure and deploy web applications.
- Creating Azure Web Apps from the gallery.
- Deploying and monitoring Azure Web Apps.
- Creating and configuring Azure Virtual Machines.

- Create and manage a storage account.
- Manage blobs and containers in a storage account.
- Create, configure and connect to a SQL Databases instance.
- Identify the implications of importing a SQL standalone database.
- Manage users, groups and subscriptions in an Azure Active Directory instance.
- Create a virtual network.

Course Outline

The course outline is as follows:

Module 1, "Overview of the Microsoft Azure Platform"

Microsoft Azure provides a collection of services that you can use as building blocks for your cloud applications.

Module 2, "Building Application Infrastructure in Azure"

Although many Microsoft Azure services use virtual machines, sometimes your application might have a unique need where it requires a virtual machine that is completely unmanaged. Azure provides networking, backup, and virtualization services as part of its Infrastructure-as-a-Service (IaaS) offering.

Module 3, "Hosting Web Applications on the Azure Platform"

This module provides an overview of the Azure Web Apps service.

Module 4, "Storing SQL Data in Azure"

Dynamic web applications must store the data that is being managed and manipulated by end users. ASP.NET technologies such as ADO.NET and Entity Framework provide a way for accessing data in SQL Server. In the cloud, the Microsoft Azure platform provides a database as a service offering that allows developers to use SQL in the same way as they would in an on-premises location.

Module 5, "Designing Cloud Applications for Resiliency"

As a developer, you should keep in mind certain considerations while designing applications for the cloud. Although there are many platform improvements available in the ASP.NET ecosystem, you need to rethink the way you design your applications, and the patterns that are used, with respect to the scalability and reliability metrics present for the cloud applications. Lesson 1, "Application Design Practices for Highly Available Applications," discusses some of the considerations that are needed when you design applications that are hosted in the cloud such that they result in minimal downtime.

Module 6, "Storing Unstructured Data in Azure"

Many new application workloads require new databases that offer scale and flexibility far beyond the capabilities of a traditional relational database. In Azure, there is a wide variety of NoSQL database services available for applications to store unstructured data in a flexible, schema-free and scalable fashion.

Module 7, "Storing and Consuming Files from Azure Storage"

When you want to scale to different cloud instances, storing files to a local disk becomes a difficult process to maintain and eventually an unreliable method of storage. Azure provides a Blob storage mechanism that not only offers high performance but also supports integration to Microsoft Azure Content Delivery Network (CDN) for low latency downloads.

Module 8, "Designing a Communication Strategy using Queues and Service Bus"

With web applications presenting content and worker roles processing the logic, there needs to be a mechanism that facilitates the communication between these different entities. Microsoft Azure provides two queuing mechanisms that you can use for this purpose.

Module 9, "Automating Integration with Azure Resources"

Although you can manage most of the Azure services by using both of the Azure portals or Microsoft Visual Studio, you can use scripting to completely automate the management of the same resources. This module will look at automating the lifecycle of the services by using client libraries, Windows PowerShell, REST, and the Resource Manager.

Module 10, "DevOps in Azure"

Although you can deploy your cloud applications manually, it is in your best interest to begin automating cloud-based deployments. Automation creates many benefits including the ability to trace past actions, easier repetition of deployment tasks and reduced possibility of human error.

Module 11, "Securing Azure Web Applications"

Just like on-premises applications, applications in the cloud need streamlined security mechanisms that are flexible. Azure Active Directory is an identity provider that can provide identity and access functionality for your custom applications or SaaS applications.

Course Materials

The following materials are included with your kit:

- **Course Handbook** is a succinct classroom learning guide that provides the critical technical information in a crisp, tightly focused format, which is essential for an effective in-class learning experience.

You may be accessing either a printed course handbook or digital courseware material via the Skillpipe reader by Arvato. Your Microsoft Certified Trainer will provide specific details, but both printed and digital versions contain the following:

  - **Lessons** guide you through the learning objectives and provide the key points that are critical to the success of the in-class learning experience.
  - **Labs** provide a real-world, hands-on platform for you to apply the knowledge and skills learned in the module.
  - **Module Reviews and Takeaways** sections provide on-the-job reference material to boost knowledge and skills retention.
  - **Lab Answer Keys** provide step-by-step lab solution guidance.

 **Additional Reading: Course Companion Content on the https://www.microsoft.com/learning/companion-moc.aspx website.** This is searchable, easy-to-browse digital content with integrated premium online resources that supplement the Course Handbook.

- **Modules** include companion content, such as questions and answers, detailed demonstrations steps, and additional reading links for each lesson. Additionally, modules include Lab Review questions and answers and Module Reviews and Takeaways sections, which contain the review questions and answers, best practices, common issues and troubleshooting tips with answers, and real-world issues and scenarios with answers.
- **Resources** include well-categorized additional resources that give you immediate access to the current premium content on TechNet, MSDN, and Microsoft Press.

 **Additional Reading: Student Course files** includes the Allfiles.exe, a self-extracting executable file that contains all required files for the labs and demonstrations hosted on GitHub.com.

- **Course Evaluation**. At the end of the course, you will have the opportunity to complete an online evaluation to provide feedback on the course, training facility, and instructor.
  - To provide additional comments or feedback on the course, send an email to mcspprt@microsoft.com. To inquire about the Microsoft Certification Program, send an e mail to mcphelp@microsoft.com.

Virtual Machine Environment

This section provides the information for setting up the classroom environment to support the business scenario of the course.

Virtual Machine Configuration

In this course, you will use Microsoft Hyper-V to perform the labs.

Note: At the end of each lab, you must revert the virtual machines to a snapshot. You can find the instructions for this procedure at the end of each lab.

The following table shows the role of each virtual machine that is used in this course:

Virtual machine	Role
20532-SEA-DEV	A client computer that is running Windows 10

Software Configuration

The following software is installed on the virtual machine:

- Visual Studio Code
  - Azure Resource Manager Tools
  - Azure CLI Tools
- Azure PowerShell
- Azure CLI
- Microsoft Azure Storage Explorer

The following Windows features are enabled on the virtual machine:

- Windows Subsytem for Linux

The following Windows Store apps are installed on the virtual machine:

- Ubuntu Linux Shell

Course Files

The files associated with the labs in this course are located in the <install_folder>\Labfiles\LabXX folder on the student computers.

Classroom Setup

Each classroom computer will have the same virtual machine configured in the same way.

You may be accessing the lab virtual machines either in a hosted online environment with a web browser, or by using Hyper-V on a local machine. The labs and virtual machines are the same in both scenarios; however, there may be some slight variations because of hosting requirements. Any discrepancies will be pointed out in the Lab Notes on the hosted lab platform.

Your Microsoft Certified Trainer will provide details about your specific lab environment.

Course Hardware Level

To ensure a satisfactory student experience, Microsoft Learning requires a minimum equipment configuration for trainer and student computers in all Microsoft Learning Partner classrooms in which Official Microsoft Learning Product courseware is taught.

The instructor and student computers must meet the following hardware requirements:

- Processor: 2.8 GHz 64-bit processor (multi-core) or better
    - AMD:
        - AMD Virtualization (AMD-V)
        - Second Level Address Translation (SLAT)—nested page tables (NPT)
        - Hardware-enforced Data Execution Prevention (DEP) must be available and enabled (NX Bit)
        - Supports TPM 2.0 or greater
    - Intel:
        - Intel Virtualization Technology (Intel VT)
        - Supports Second Level Address Translation (SLAT)—Extended Page Table (EPT)
        - Hardware-enforced Data Execution Prevention (DEP) must be available and enabled (XD bit)
        - Supports TPM 2.0 or greater
- Hard Disk: 500GB SSD System Drive
- RAM: 32 GB minimum
- Network adapter
- Monitor: Dual monitors supporting 1440 x 900 minimum resolution
- Mouse or compatible pointing device
- Sound card with headsets

In addition, the instructor computer must be connected to a projection display device that supports a minimum resolution of WXGA 1280 x 800 (16x10) pixels, 16-bit colors.

Module 1

Overview of the Microsoft Azure Platform

Contents:

Module Overview

Microsoft Azure provides a collection of services that you can use as building blocks for your cloud applications. Lesson 1, "Azure Services," provides a recap of the services that you might have worked with when using the Microsoft Azure platform in the past. Lesson 2, "Azure Portal," describes the Azure portal that is available for managing Azure subscriptions and services.

Objectives

After completing this module, you will be able to:

- Describe some of the common Azure services.
- Describe features of the Azure Portal.

Lesson 1
Azure Services

This lesson describes some of the common services and features, used by many entry-level Azure developers and IT professionals, which are available in Azure. Although not comprehensive, this feature list represents many of the services that you might have used prior to taking this course.

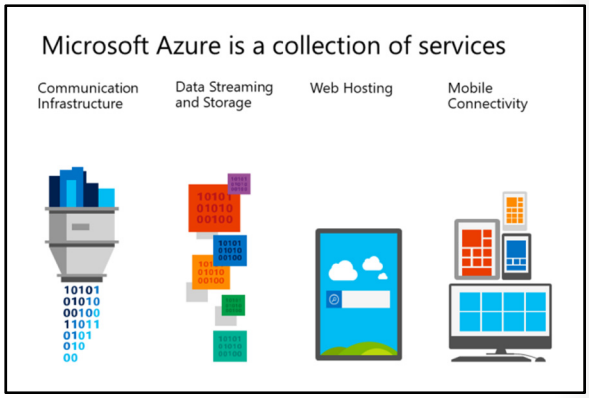
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the following Azure services:
  - Web Apps
  - Virtual Machines
  - Storage
  - SQL Database
  - Virtual Network

Services Overview

The Azure platform is a collection of services that allows you to host existing workloads, use managed services instead of a workload, and create greenfield workloads. In the majority of projects, you might use only a subset of the available services in the Azure platform. It is very typical to find varying levels of experience amongst your peers. In this course, you will learn about the most common services that you can use in new development projects that will be hosted on the Azure platform.



In your development projects or workloads, you can choose to use as many available services as needed to meet your requirements. The following sections describe the common Azure services.

Physical Infrastructure Alternatives

Services such as Virtual Machines and Virtual Network help in emulating the existing infrastructure in your datacenter. You can use these services in the *lift and shift* scenarios where you want to migrate existing virtual machines from an on-premises datacenter to Azure. You can use Virtual Machines and Virtual Network in scenarios where you want to have maximum control over your operating system environment and the configuration settings of a cloud application.

Communication Infrastructure

Cloud applications are typically modular in nature and require a managed communication infrastructure, such as Service Bus, to connect the various application modules. Service Bus provides features such as event subscription and publishing, mobile notification infrastructure, WCF service relay, and a managed queue service.

Data Streaming and Storage

The managed Storage service can handle a wide range of file persistence and manipulation scenarios. The Blob and Azure file services help you to easily save and modify files either by using a client library or the SMB 2.1 protocol.

Web Hosting

For low-friction development and deployment, the Web Apps service provides a web hosting platform for your applications that integrate with File Transfer Protocol (FTP), source control providers, or the Web Deploy protocol.

Mobile Connectivity

Mobile Apps is a fully managed back-end service solution for client devices. Together with SQL Databases, Mobile Apps can help you design your data schema while you scale up to meet the demand from your users.

Web Apps

Web Apps provides platform for your web applications with many features to reduce the friction of deployment. A high degree of compatibility with Internet Information Server (IIS) Web Apps allows you to configure your web applications by using familiar IIS configuration settings. Because the platform is highly managed, you can scale up your web applications automatically with load balancing, which is also configured automatically.

Reference Link:
https://docs.microsoft.com/azure/app-service/app-service-web-overview

The Web Apps service allows you to use ASP.NET, Java, PHP, Node.js, or Python as an application framework for your web applications out of the box. You can also use popular CMS solutions with custom templates and markup languages, such as Drupal, Joomla, DotNetNuke, and Umbraco, with the platform for your web solutions.

Deployment

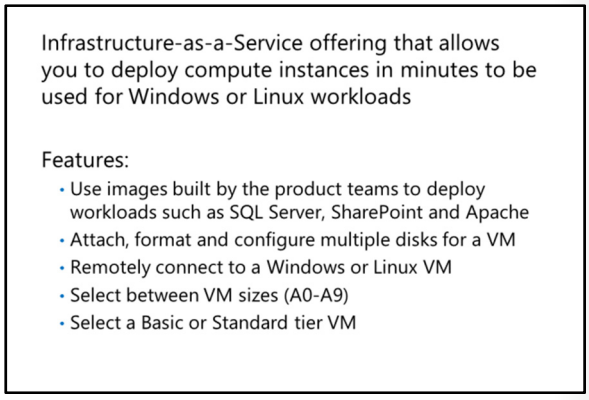
Along with the current synchronous deployment methods, Web Apps also supports continuous deployment from a source control provider. You can use a source control provider such as GitHub, Bitbucket, Codeplex, or Microsoft Azure Visual Studio Online as the source for a continuous deployment build.

No Ops

The Web Apps environment is fully managed, which allows you to have a scalable web hosting platform that is resilient and highly available without having to specifically manage each Web App instance at a low level. This also allows you to scale up your web application by simply configuring a set of values.

Virtual Machines

Virtual machines provide compute on demand for your application workloads with a high degree of compatibility with existing virtualization workloads. By using the standardized .vhd format from Hyper-V disks, infrastructure administrators can easily on ramp existing Hyper-V workloads to Azure.



Workloads

Virtual machines support a list of common Microsoft application workloads. Software platforms such as Microsoft SharePoint, SQL Server, Dynamics GP, and BizTalk Server are fully supported using the infrastructure hosted in Azure. By connecting Virtual Machines to your on-premises network, you can enable disaster-recovery and high-availability solutions such as SQL Server AlwaysOn.

Templates

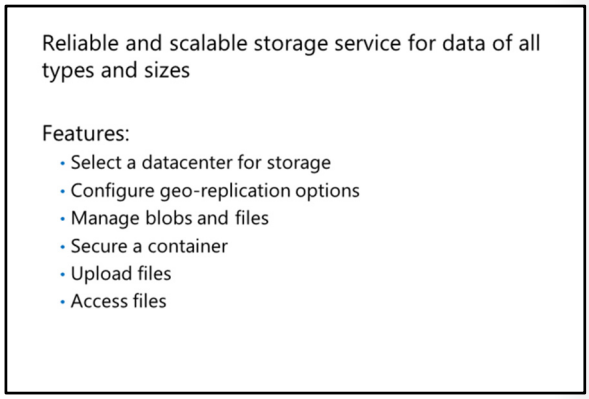
You can create the majority of supported workloads in Azure by using pre-built templates on the portal. Microsoft and third-party images are available for both Linux and Microsoft-supported application workloads. Standard operating system images for Windows and Linux are also available for new projects.

Choice in Sizing

You can configure virtual machines with a variety of options for CPU, memory, and IOPS, commonly referred to as VM sizes. The Basic and Standard sizes are available for the most common application workloads. Larger standard sizes are also available for memory intensive workloads. D-size virtual machines are available with a faster processor and a local solid-state drive (SSD). G-size virtual machines are available for workloads that require massive compute or memory resources.

Storage

Storage is a managed service that allows you to store data that can be used throughout your cloud applications. The data that is stored can include loosely structured entities, queue messages, and various files. You can access this data by using client libraries, URLs, or the REST API. You can store and manage files by using files shares.



Blobs

Blobs are managed files that can be persisted and accessed by using URLs, the REST API, or a client library. You can use containers to logically group blobs. Blobs are used throughout service instances, such as Virtual Machines that uses blobs to store virtual hard disks.

Queues

The Queue service provides a simple managed interface to push messages into a queue and consume the same messages. Messages are stored as serialized strings. They can be stored or retrieved in a First-In First-Out manner by using logical queue actions, such as enqueue, dequeue, and peek.

Tables

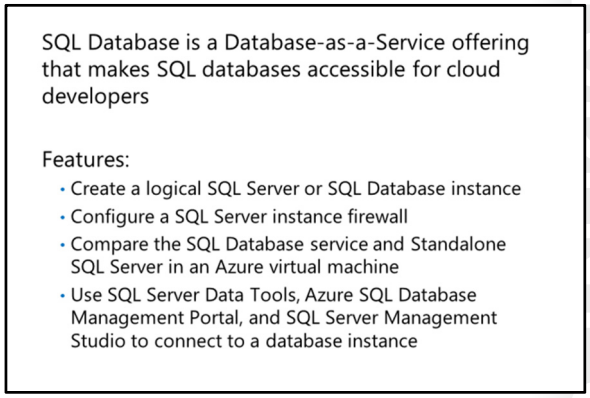
The Table service is a NoSQL store that allows you to store loosely structured sets of entities that can be persisted en masse and retrieved with exception efficiency. By implementing common sharding concepts, such as partitions and hash indexes, you can use tables to store large quantities of data for your application.

Files

The Azure Files service provides an SMB 2.1 protocol file share that you can use when you migrate existing application workloads from on-premises to Azure. The files that are persisted by using this technology can also be accessed by using the REST API or client libraries.

SQL Database

SQL Database is a managed Database-as-a-Service platform that you can use to host your SQL objects. It is highly compatible with existing data tools and therefore it provides you with similar management experience that you will have with SQL Server Standalone.



Compatibility

Many SQL features and objects can be used with SQL Database. SQL Server 2014 provides tooling that you can use to analyze and migrate your databases from an on-premises server to Azure.

Data Tools

You can use many existing data tools such as SQL Server Data Tools for Microsoft Visual Studio and SQL Server Management Studio with SQL Database. Other tools and application frameworks can use connection strings to connect to the databases that are hosted in Azure. You can provide the connection strings for SQL Database instances directly on either portal. SQL Database also provides a custom portal for managing databases.

Scalability

Elastic Scale can be used with a SQL Database instance to automatically create and manage defined partitions for applications with dynamic or large database workloads. Applications that can benefit from this feature range from large data stores to multitenant Software-as-a-Service (SaaS) services.

Virtual Networks

Virtual networks are a security boundary that you can create so that service instances can communicate with each other privately. Virtual networks support instances of the Web Apps, SQL Databases, and Virtual Machines services. Through the various connectivity options, virtual networks can be connected to each other or to an existing on-premises machine or network.

Private network that is available for grouping of services and compute instances in the cloud or on premise

Features:
- Create a Virtual Network (VNET) specifying a region or affinity group
- Configure a VNET to use a DNS server
- Configure VNET subnets
- Implement a point to site connection to a VNET
- Create a virtual machine in an existing VNET

Site Connectivity

You can connect virtual networks with each other or with an existing on-premises network by using the Internet Protocol Security (IPSec) protocol. This allows a secure link between multiple networks and various hybrid cloud scenarios. VPN Devices that support this connectivity can be determined by using a list of known supported VPN devices or a list of features that are required to be supported by a device that will be used with a Site-to-Site connection.

Reference Link:
https://docs.microsoft.com/azure/vpn-gateway/vpn-gateway-about-vpn-devices

Direct Connectivity

The Secure Sockets Tunneling Protocol (SSTP) can also be used to allow an individual device or a computer to directly connect to a virtual network by using a Point-to-Site virtual private network (VPN) connection. This is useful in scenarios where an on-premises location does not have the hardware that supports IPSec or a network needs to be used by multiple remote workers.

Domain Name Service (DNS)

Azure automatically provides a DNS service for virtual machines and services in a virtual network. You can optionally provide an IP address for your own DNS server that is either hosted in Azure or connected by using one of the previously mentioned connections.

App Services

Azure provides many other services for your cloud applications. The following sections describe some of these services.

Azure provides a collection of services that you can integrate in new or existing applications to enhance their functionality

Examples:
- Azure Active Directory
- Media Services
- Mobile Services
- Automation

Azure Active Directory (Azure AD)

Azure AD is a managed identity and access service. You can use this service to provide authentication and single sign-on capability for cloud applications or any custom application. Optionally, you can sync this service with an existing Active Directory domain controller.

Media Services

Media Services is a service that allows you to encode and stream multimedia for a wide variety of customers and devices. Media Services can dynamically scale up to meet the spikes in demand for the transformation and retrieval of audio or video. You can use Jobs with Media Services to monitor the progress of the encoding operations that are in the queue.

Mobile Services

Mobile Services is a back-end service platform that you can use to store and provide application data for mobile devices. Mobile Services uses SQL Database to store the actual data and manage the schema of the data that is posted to the back-end web service.

Automation

Automation extends your management features by allowing you to use Windows PowerShell to automate common management tasks. Automation uses Windows PowerShell workflows in the same manner as Microsoft System Center. Automation also includes an extensive Runbook library of scripts that are provided by Microsoft and the open-source community.

Lesson 2
Azure Portal

Many iterations of Azure services and the web applications to manage those services are currently available.

This lesson describes the last two iterations of the Azure Portal, which you can use for configuring instances of Azure services. This lesson also provides a walkthrough of how to switch between the two Azure Portals when the functionality you require is not available in one of the portals.

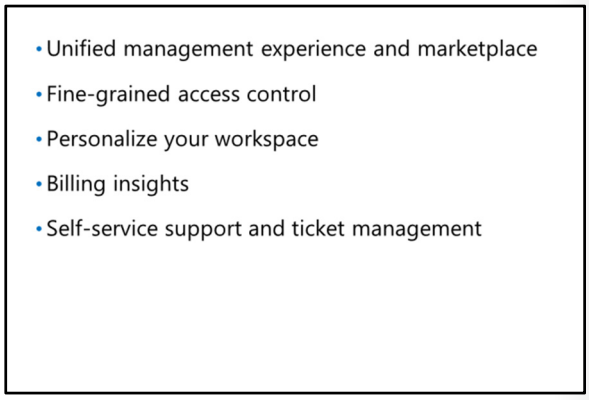
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the current Azure Portal.
- Describe the Classic Azure Portal.
- Switch between the different portals.

The Azure Portal

Azure as a platform is a collection of services that you can use to host your applications and workloads in the cloud. The Service Management REST API is a web service that receives requests to create, change, or configure services and passes the requests to the Microsoft Azure Fabric Controller. The Fabric Controller makes decisions based on these requests and utilizes the Azure Hypervisor to create new virtual machines, as necessary, in the datacenters.



The Azure Portal is one of many different interfaces that you can use to interact with the Azure platform. All the interfaces share the Service Management REST API and this API can be used by any custom application you create for managing services in the Azure platform.

The latest version of the portal was released at //build 2014. The focus of this new portal is to display more metadata about each service instance and group the services logically for monitoring and billing. Additional features are released for the new portal on a weekly basis and this rapid cadence.

Billing

Previously, you had to switch from the Classic Portal to the Account Portal to view the billing data for your services. In the Portal, you can view the up-to-date billing data for your entire subscription in the form of charts and infographics. The charts and infographics will help you understand the impact of individual service instances and service types. Charts, such as the Burn rate, also help you forecast the charges to your account in each billing period.

Billing visualizations in the Portal:

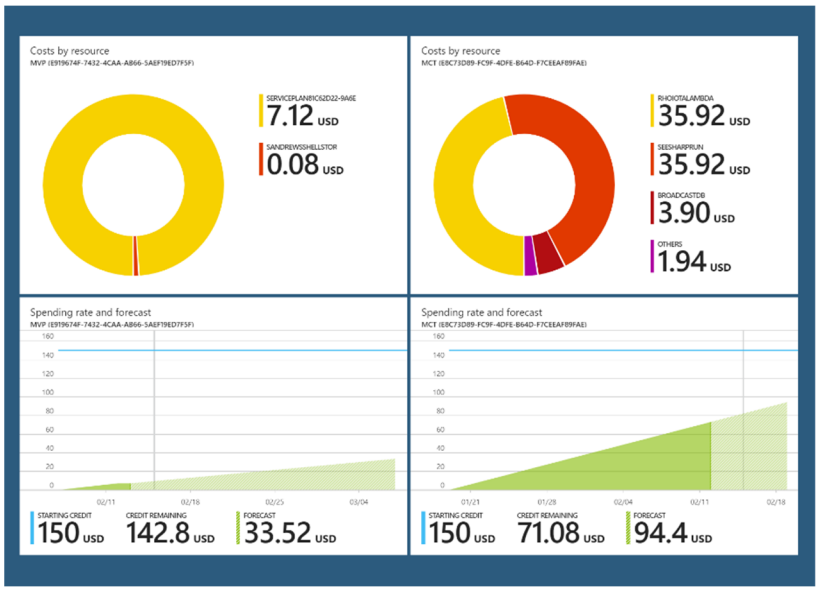


FIGURE 1.1: THE BILLING BLADE

Resource Groups

Resource groups represent a new way of logically grouping your service instances. Service instances are known as *resources* and a collection of resources can exist in a *resource group*. By using resource groups, you can view the metrics and billing data for a specific group in your subscription. Resource groups also allow your service instances to share a common lifecycle, where you can create a group with multiple resources defined or remove a group and the Resource Manager ensures that the individual resources are also removed. Resource groups are covered in depth in Module 11, "Automating Integration with Azure Resources."

Resource groups and role-based access control:

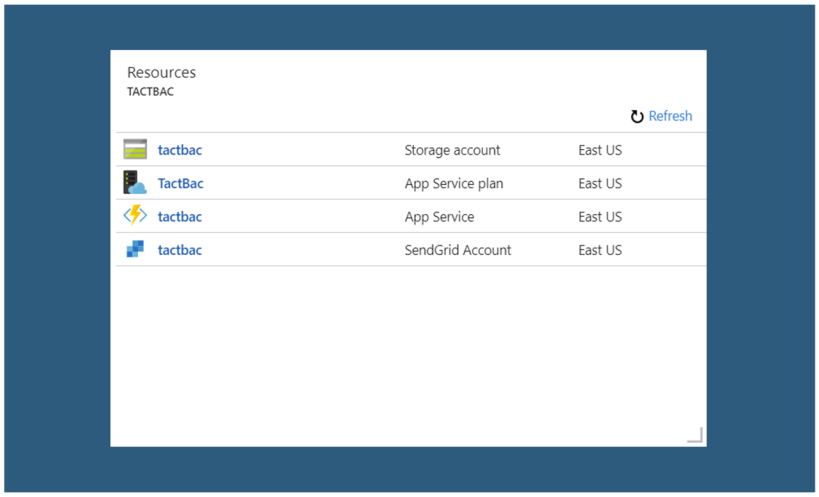


FIGURE 1.2: RESOURCE GROUP

Portal Features

Blades

- Create and Configure Cloud Resources
  - Virtual Machines, Virtual Networks, Web Apps, Mobile Apps etc.
- Monitor Workloads and Configure Alerts
  - Customizable Dashboard and blades
- Integrated DevOps Experience
  - Continuous integration and deployment and configuration management
- Integration with the Azure Marketplace
  - Easy access to Microsoft and Azure certified solutions

When you view the settings for a service instance or take an action, the details are displayed in a vertical dialog box known as a blade. You can stack these blades horizontally so that you can view the details of a service (or an action) and some of the follow-up actions without having to scroll horizontally. You can close a blade by clicking the close (X) button at the top-right corner. You can click the pin icon at the top-right corner to pin the blade to your Dashboard for future access. You can minimize the blades to place them on the left side of the screen and maximize them to fill the entire width of your screen. You can hide the labs on the top command bar to create more screen space for your tiles. Finally, you can customize the order and layout of the tiles in most blades.

The New blade to create resources in Azure:

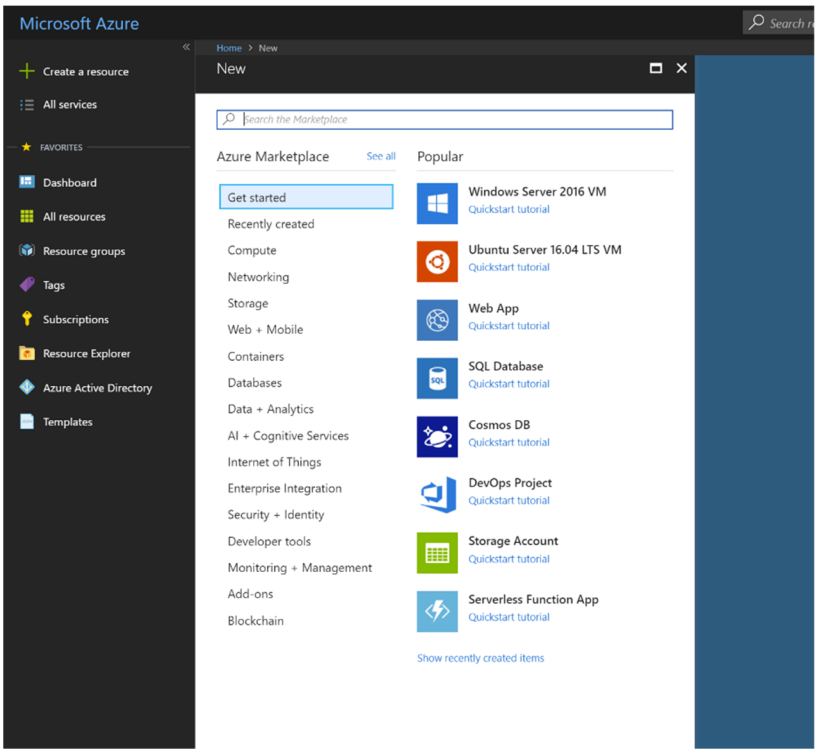


FIGURE 1.3: THE NEW BLADE

Journeys

Journeys are a collection of sequential blades that you can use to create or modify a service. For example, to create a Web App, you can go to the gallery blade, and then view the Web App gallery options blade. After you select a specific Web App template, a blade describing the template displays. If you click Create, another blade displays with basic options for the new Web App. You could drill down further and specify advanced options. Many of these advanced options might display a subsequent blade. These blades that are displayed in a horizontal sequence are known as a *journey*. In a journey if you try to close a blade,

which is to the left of another blade, without saving the changes, a message displays informing you about the loss in progress. Journeys are sometimes broken down in to *Journey Parts*, but this is not currently visualized in the user interface.

The following image depicts a journey from the "All Resources" blade to a specific "Storage Account" blade:

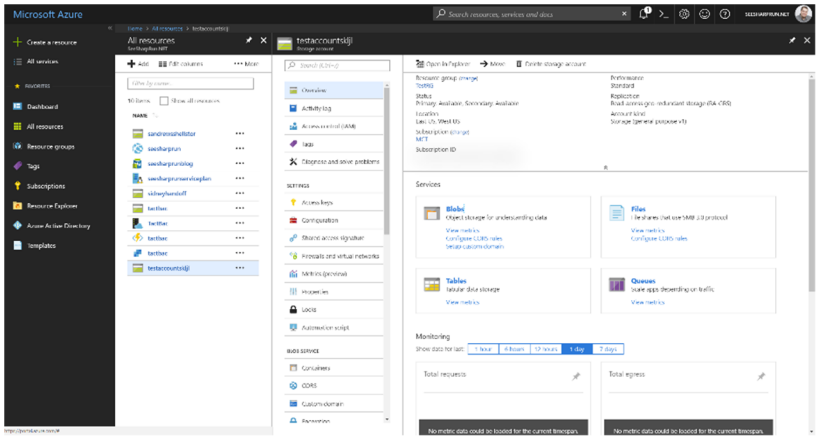


FIGURE 1.4: A JOURNEY

Dashboard

The Dashboard is the first screen you see when you log on to the Portal. It is a collection of tiles that you can reorganize, resize, and remove. As you view the blades in your subscription, you can pin them to the Dashboard so that you can return to them at a later point in time. You can also resize the tiles to emphasize specific data.

Most common resize options for the tiles on the Dashboard:

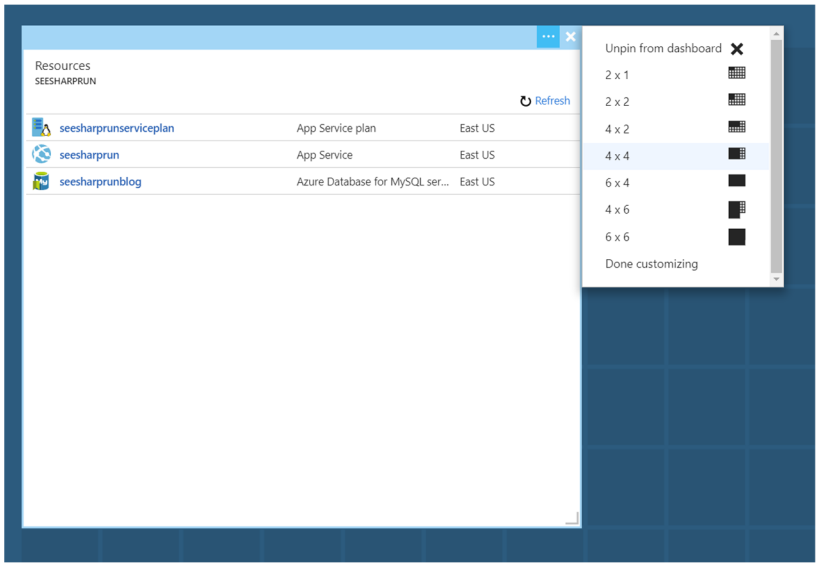


FIGURE 1.5: RESIZE OPTIONS

Many tiles show different metrics and icons depending on their size. For example, the Service health tile displays a world map in the largest size, but in the smallest size it displays only a count of the services that are healthy. By default, when you create a new instance of a service, the option to add a tile to the Dashboard for that instance is selected.

You can opt to resize a tile by dragging the bottom-right corner of the tile:

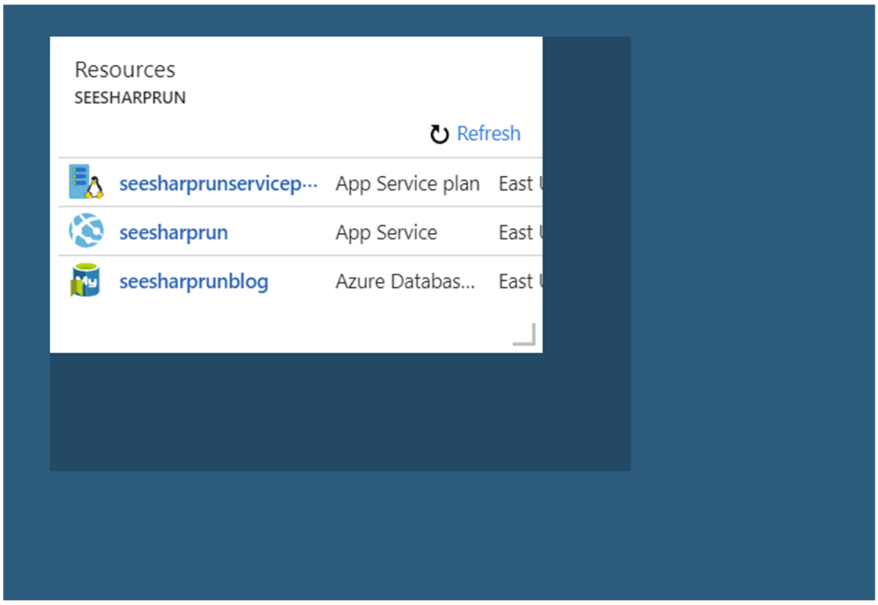


FIGURE 1.6: RESIZE DRAGGING

A common Dashboard showing a collection of pinned tiles:

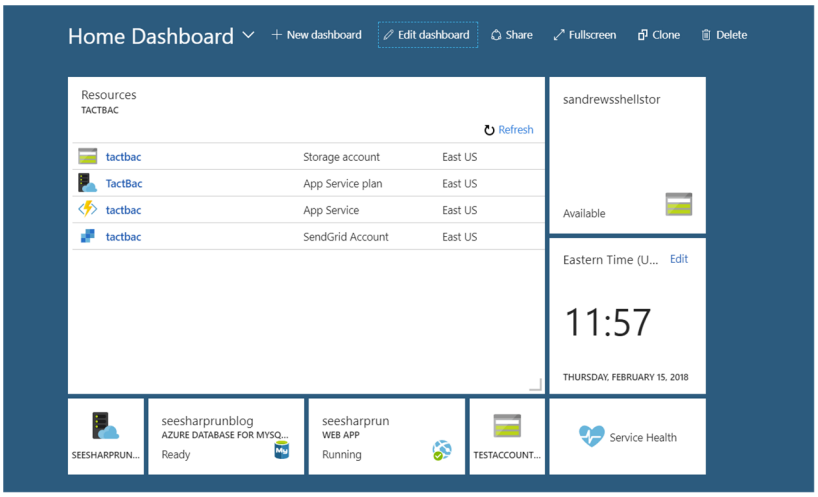


FIGURE 1.7: DASHBOARD WITH CUSTOM TILES

Demonstration: Using the Azure Portal

Note: To view the latest demo steps, visit the GitHub repository for the course.

For this demonstration, you will use the available host machine. Before you begin this demonstration, you must complete the following step:

- Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Lab: Exploring the Azure Portal

Scenario

You are designated by your team as the individual who will explore the Azure Portal and then train the other team members on how to use the portal. You decided to customize a few features of the portal and create a new service instance.

Objectives

After you complete this lab, you will be able to:

- Sign in to the Azure Portal.
- Customize your Dashboard.
- Identify a blade.
- Identify a journey.
- Identify a journey part.
- Close a journey without persisting your changes.

Lab Setup

Estimated Time: 15 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Signing in to the Azure Portal

Exercise 2: Customizing the Azure Portal

Module Review and Takeaways

In this module, you learned about Azure as a platform and its capabilities. You also previewed the tools and features that you can use to manage an Azure subscription.

Best Practice

Many of the services that are covered in this module are pre-requisites for the course. If you never worked with any of services that are mentioned in this module, please take time to review these services before completing the subsequent modules. This will ensure that you are prepared for the later modules in this course.

Review Question

Question: You are building an iOS and Android application. Your application will need a back-end web service, and you decided to host the service in Azure. What services can you use to accomplish this task?

Module 2

Building Application Infrastructure in Azure

Contents:

Module Overview

Although many Microsoft Azure services use virtual machines, sometimes your application might have a unique need where it requires a virtual machine that is completely unmanaged. Azure provides networking, backup, and virtualization services as part of its Infrastructure-as-a-Service (IaaS) offering. Lesson 1, "Azure Virtual Machines," introduces the Virtual Machines service and describes the options that you can use for creating a virtual machine. Lesson 2, "Azure Virtual Machine Workloads," provides details on the types of workloads that you can deploy to a virtual machine. Lesson 3, "Migrating Azure Virtual Machine Instances," describes the options for migrating virtual machines to and from Azure. Lesson 4, "Highly Available Azure Virtual Machines," reviews the options and features that must be considered when designing your Virtual Machine instances for high availability scenarios. Lesson 5, "Virtual Machine Configuration Management," describes the common methods for managing and duplicating the configuration for virtual machines. Lesson 6, "Customizing Azure Virtual Machine Networking," reviews the options for managing inbound and outbound connection rules for your virtual machine. Lesson 7, "Virtual Machine Scale Sets," introduces the VMSS service and describes how it can be used to automatically provision virtual machines for autoscale scenarios.

Objectives

After completing this module, you will be able to:

- Describe the Virtual Machines service in Azure.
- Deploy a Linux or Microsoft workload to a virtual machine.
- Import virtual hard disks to Azure.
- Monitor virtual machine endpoints.

Lesson 1
Azure Virtual Machines

The Virtual Machines service in Azure provides quick compute that can be scaled up or out and completely customized. The Azure Management Portal provides a large collection of templates that makes it very easy for you to get started with a popular server operating system.

This lesson describes the Virtual Machines service in Azure and provides the details on some of its unique features.

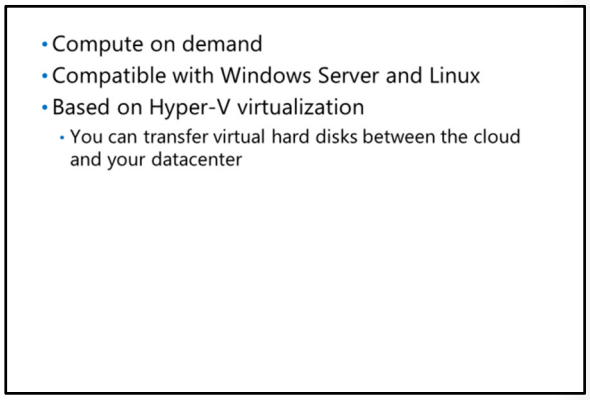
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Virtual Machines service.
- Describe the prebuilt and custom image options.

Virtual Machines Overview

Virtual Machines provides a highly flexible compute-on-demand option for running your application workloads. Azure Virtual Machines offer a variety of virtual machines sizes, compatibility with Windows and Linux and deep compatibility with Hyper-V fixed-size virtual hard disks. Virtual machines hosted in Azure can host a wide variety of workloads.



Scale and Availability

Virtual machines in Azure can be grouped for high scalability and availability. By placing multiple virtual machines in a cloud service, you can stand up multiple instances of any tier of your application. For example, you can host a web application on four Windows Server 2012–based virtual machines that have Internet Information Services (IIS)enabled. You can place the virtual machines in an availability set so that at least one of the virtual machines will be available at all times. However, according to the Azure uptime Service Level Agreement (SLA), you must have an availability set with at least two instances of your virtual machine. If you add multiple instances of your virtual machine to an availability set, you should consider configuring autoscale. Autoscale allows you to start and stop virtual machines to meet the demand of your application. In the case of virtual machines, you will create the maximum number of instances that you think your application will need, and then enable autoscale with a metric defined which would enable the scale action. Autoscale and availability sets are discussed further in Module 10, Managing Infrastructure in Azure.

Hyper-V

Virtual machines in Azure use the well-known Hyper-V virtual hard disk format (.vhd) for their hard drives. Because Azure uses the .vhd format, you can simply upload fixed-size virtual hard disk files from your existing infrastructure to Azure. You can also download virtual hard disk files from Azure to your datacenter.

Using Images to Construct Virtual Machines

The Management Portal provides many images and scripting tools that help you to create new virtual machines in Azure. The template images that are available in the portal are created and fully supported by either Microsoft or an authorized third-party. You can use these images as a base for creating a greenfield project in Azure or for migrating an existing workload to Azure.

- Many images are already provided by Microsoft:
  - Microsoft SQL Server
  - Microsoft SharePoint
  - OpenSUSE
  - Microsoft BizTalk Server
- Your Azure subscription might have some custom images
  - For example, MSDN subscriptions come with Windows 7 and Windows 8.1 images with Microsoft Visual Studio preinstalled

Microsoft Images

Many of the Microsoft server software are already supported in the Azure virtual machine environment. Common Microsoft workloads that are supported include:

- Microsoft BizTalk Server 2013
- Microsoft Dynamics AX, Microsoft Dynamics GP and Microsoft Dynamics NAV
- Microsoft Project Server 2013
- Microsoft SharePoint Server 2010 and Microsoft SharePoint Server 2013
- Microsoft System Center 2012 Service Pack 1
- Microsoft SQL Server 2008, Microsoft SQL Server 2012, and Microsoft SQL Server 2014
- Microsoft Team Foundation Server 2012

All the supported Microsoft workloads are not available as template images. For example, SQL Server 2008 is a supported workload, but a gallery image is available only for Microsoft SQL Server 2008 R2.

A virtual machine can be created using the Microsoft SQL Server 2012 SP1 gallery image in the Management Portal:

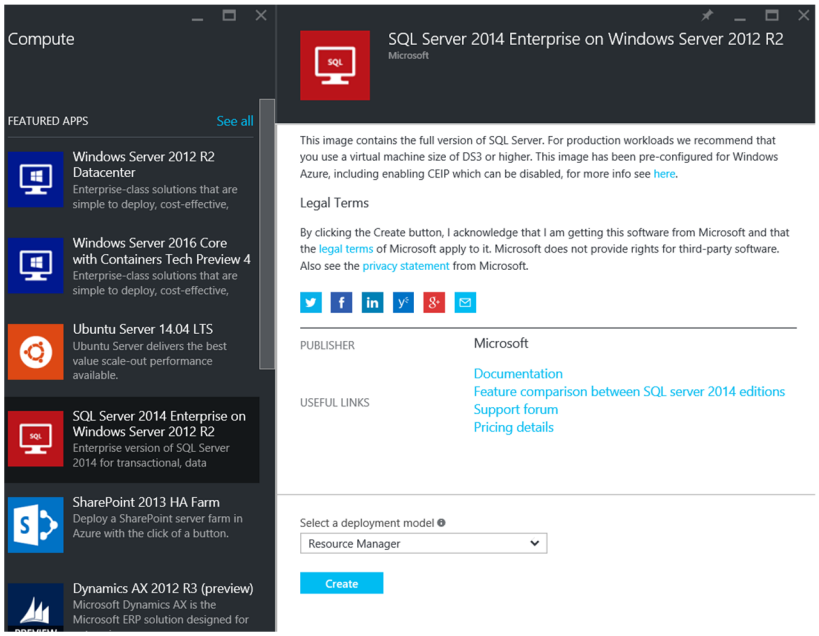


FIGURE 2.1: SQL SERVER 2012 SP1 GALLERY TEMPLATE

Open Source and Third-Party Images

Microsoft and third-party providers have various open-source images including:

- CoreOS
- Ubuntu
- openSUSE
- OpenLogic

You can also create your own custom Linux–based virtual machines and upload them to Azure as .vhd files.

Creating an Ubuntu Server 14–based virtual machine from a gallery image in the Azure Preview Portal:

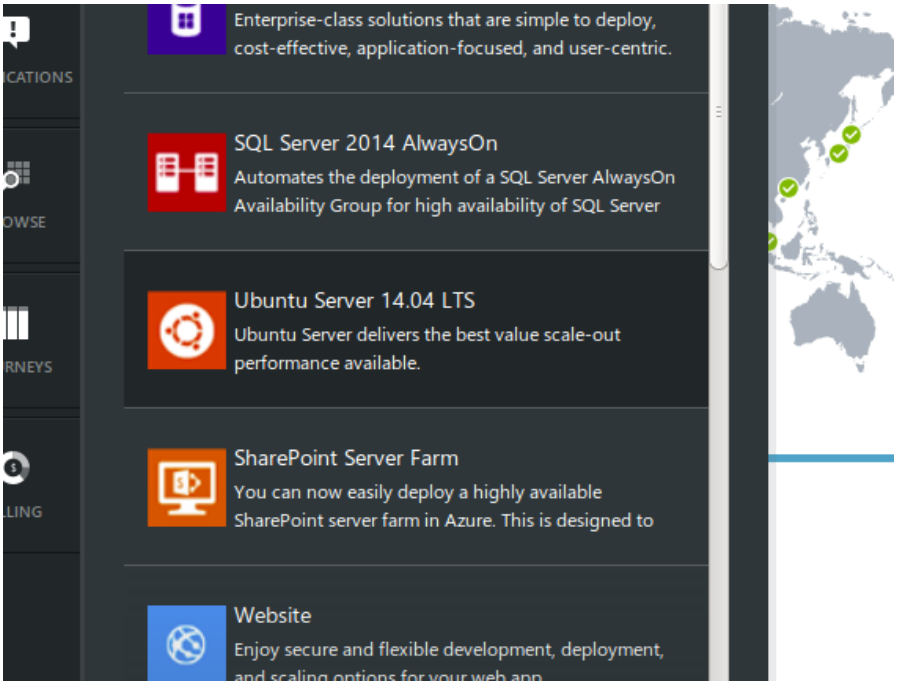


FIGURE 2.2: UBUNTU SERVER 14 GALLERY TEMPLATE

Custom Images

You can create custom images from an existing virtual machine. You use the custom images to create duplicates of your virtual machine in scenarios where you need multiple instances of the virtual machine to have a similar configuration. Custom images can either be generalized, so that you can clone as many copies as you want, or specialized so that you can create a checkpoint of a virtual machine to maintain its state at a certain point in time.

Generalized virtual machine images are operating system images that are de-provisioned (Sysprep in Windows or waagent in Linux) so that you can provision them after they are cloned to other virtual hard disks.

Note: In Azure, when you capture a generalized virtual machine image, the image will delete the virtual machine without affecting the disk. You can then use this image to create as many duplicate virtual machines as you need. You must stop a virtual machine before capturing a generalized image.

Specialized virtual machine images are a snapshot of a running virtual machine, which you can use later to return the virtual machine to that point in time.

Note: In Azure, you can capture a specialized virtual machine image from a virtual machine that is running or stopped. This will not affect the existing virtual machine. You can create a new virtual machine by using the specialized image that need not be provisioned again.

Demonstration: Creating a Virtual Machine

In this demonstration, you will learn how to:

- Create a virtual machine by using the Preview Portal.
- Use Remote Desktop to access the virtual machine.

Lesson 2
Azure Virtual Machine Workloads

You can deploy your existing application workloads to virtual machines in Azure that are running either on the Windows or Linux operating system.

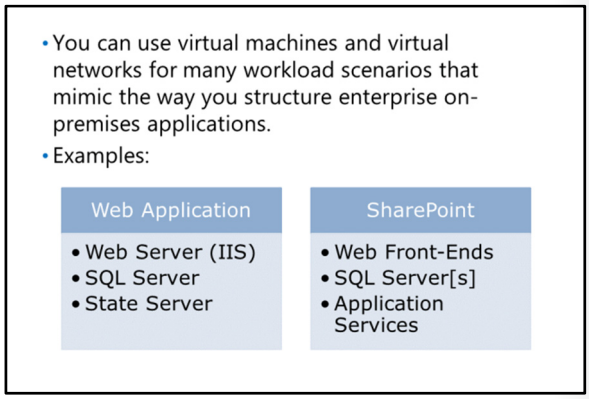
This lesson describes some of the considerations for deploying application workloads to the virtual machines in Azure.

Lesson Objectives

After completing this lesson, you will be able to describe the key considerations for deploying Windows or Linux workloads to Azure.

Windows Workloads

Many common Microsoft workloads (server software) are already supported in Azure. Most of these workloads also include corresponding virtual machine templates in the Azure portals. All software that is installed on an Azure virtual machine must have the necessary licenses. However, if you use a gallery template the cost of the license is already included in the cost of the virtual machine.



Note: Virtual Machines using the Windows Server template include the cost of the Windows Server license. Virtual Machines created using a SQL Server or other Microsoft server software template are typically priced higher than virtual machines with just Windows Server installed. This is because the cost of the additional software license is added to the cost of the Windows Server license.

If you start with an operating system gallery template, and then install the server software, you must immediately license the server software by using license migration.

Supplemental guidance articles are available, which will help you plan for the installation of many common Microsoft software workloads on Azure Virtual Machines.

Reference Link: https://docs.microsoft.com/office365/enterprise/microsoft-azure-architectures-for-sharepoint-2013

Reference Link: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-server-iaas-overview

Linux Workloads

Many common Linux distributions are already available on the Azure platform with gallery images. Including the images in VM Depot, there are enough images in the Azure platform to run the most popular Linux workloads.

- You can use the Linux–based virtual machines that are available in Azure enable to deploy common Linux workloads
- Examples:
  - Apache Lucene
  - LAMP (Linux, Apache, MySQL, PHP)
  - Couchbase (distributed)
  - Drupal
  - Docker
  - Chef or Puppet
  - Docker

The Azure platform provides a common interface for managing virtual machines, irrespective of whether they run on Linux or Windows. Many common features, such as capturing an image, attaching a disk, and stopping a virtual machine, use common buttons in either portal and common actions in the Cross-Platform Command-Line Interface. This enables you to manage your Linux–based virtual machines in the same manner as you manage your Windows–based virtual machines.

After provisioning the virtual machine, virtual machine extensions are used to configure the Linux–based virtual machine so that it can be accessed by using Secure Shell (SSH). You can then use tools such as PuTTY to access your new Linux–based virtual machine.

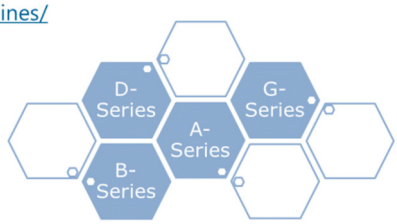
Reference Link: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows

Reference Link: https://putty.org/

Virtual Machine Sizes

Virtual machines can be provisioned in a variety of sizes. These sizes provide a variety of options to customize the performance of your virtual machine such that it is adequate for your workload and also results in a cost-effective subscription.

http://azure.microsoft.com/pricing/details/virtual-machines/



Note: The maximum input/output operations per second (IOPS) is displayed in the following format:
*<Number of disks>*x*<Maximum IOPS per disk>*
For example, 4x500 indicates that you can have a maximum of 500 IOPS per disk, for up to 4 disks.

Basic Tier A-Series

Basic tier A-series virtual machines provide a cost-effective method of hosting workloads that require only a single instance of a virtual machine without the autoscaling or load-balancing function. The below table shows the differences between the various Basic A-Series sizes.

Size	Cores	Memory	Max IOPS
A0	Shared	768 megabyte (MB)	1x300
A1	1	1.75 gigabyte (GB)	2x300
A2	2	3.5 GB	4x300
A3	4	7 GB	8x300
A4	8	14 GB	16x300

Standard Tier A-Series

Standard tier A-series virtual machines are the most common virtual machines that are used in the majority of workloads. Standard virtual machines support load balancing and autoscale. This table shows the differences between the most common sizes.

Size	Cores	Memory	Max IOPS
A0	Shared	768 MB	1x500
A1	1	1.75 GB	2x500
A2	2	3.5 GB	4x500
A3	4	7 GB	8x500
A4	8	14 GB	16x500
A5	2	14 GB	4x500
A6	4	28 GB	8x500
A7	8	56 GB	16x500
A8	8	56 GB	16x500
A9	16	112 GB	16x500

Standard Tier D-Series

Standard tier D-series virtual machines include a solid-state drive (SSD) for the temporary disk along with a faster processor and increased memory for each core.

Size	Cores	Memory	Temporary disk size (SSD)	Max IOPS
D1	1	3.5 GB	50 GB	1x500
D2	1	7 GB	100 GB	2x500
D3	2	14 GB	200 GB	4x500
D4	4	28 GB	400 GB	16x500
D11	8	14 GB	100 GB	4x500
D12	2	28 GB	200 GB	8X500
D13	4	56 GB	400 GB	16x500
D14	8	112 GB	800 GB	32x500

Standard Tier G-Series

Standard tier G-series virtual machines include the highest performing CPUs that are available in the Azure platform.

Size	Cores	Memory	Temporary disk size (SSD)
G1	2	3.5 GB	406 GB
G2	4	7 GB	812 GB
G3	8	14 GB	1,630 GB
G4	16	28 GB	3,250 GB
G5	32	14 GB	6,500 GB

Lesson 3
Migrating Azure Virtual Machine Instances

Azure provides infrastructure options that make it easy to either extend your existing datacenter or create a new environment in the cloud. With services such as Networking, Backup, Site Recovery, and Virtual Machines, Azure has the necessary services to deploy your existing complex production applications.

This lesson describes the various methods that you can use to migrate your virtual machines to Azure.

Lesson Objectives

After completing this lesson, you will be able to:

- Migrate virtual machines to Azure.
- Describe the Azure Backup service.
- Describe the Hyper-V Recovery Manager service.

Migrating Virtual Machines to Azure

Since Azure Virtual Machines natively support the Hyper-V format, you can easily migrate them from your existing Hyper-V hosts to Azure or vice versa. There are also various other options that you can use to migrate your virtual machines, analyze your existing workloads, and check the compatibility of the virtual machines with Azure.

- If you are already using Hyper-V, you can migrate your virtual machines to Azure by following these steps:
  - Ensure that your virtual machines use the (Generation 1) .vhd format and not the extended (Generation 2) .vhdx format
  - Ensure that the virtual machines are of fixed size
  - Use Windows PowerShell or third-party tools to upload the .vhd files to a storage account
- From the uploaded virtual hard disks, you can create virtual machines in Azure.

Disk Mobility

Because Azure and Hyper-V support a common .vhd format, you can easily upload and download virtual hard disks by using a third-party storage explorer or automation scripts. In Windows PowerShell, you can use the following activities to add and save virtual machines:

- Add-AzureVhd
- Save-AzureVhd

In Cross-Platform Command-Line Interface, you can use the following commands to create and upload virtual machine disks to Azure:

- vm disk create
- vm disk upload

In most cases, you must use scripts to upload the .vhd files to a storage account because you cannot upload them by using any of the Azure portals. The .vhd files must be Generation 1 Hyper-V disks and they must of a fixed size. Generation 2 Hyper-V disks (.vhdx) are currently not supported.

Migration Accelerator

The Migration Accelerator is a tool that you can use to analyze your existing application workloads and then perform a full migration of your virtual machine along with the network and endpoint configuration. The migration is performed by using a series of agents and configuration or process servers. You can use the Migration Accelerator to analyze the existing multi-tier application workloads that are running on Windows Server 2008 R2 for an automated migration. Migration to Azure can happen automatically without affecting your existing infrastructure and the target site can also be tested extensively before cutover occurs. Migration Accelerator supports existing workloads on physical machines, VMWare, Hyper-V, Amazon Web Services, and various other platforms.

Reference Link: https://docs.microsoft.com/azure/site-recovery/migrate-overview

You can use Migration Accelerator to discover existing workloads, perform a migration, and then perform a cutover:

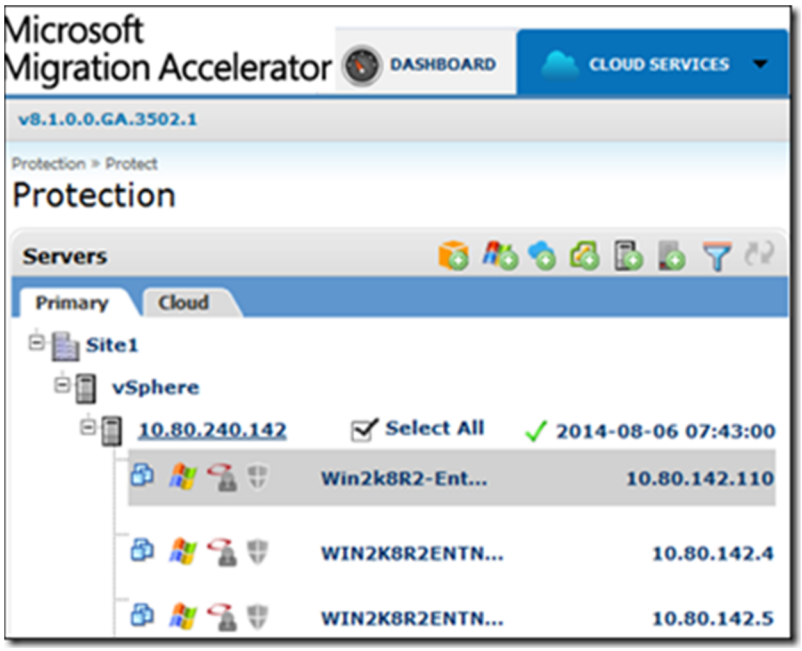


FIGURE 2.3: MIGRATION ACCELERATOR DASHBOARD

Readiness Assessment

The Virtual Machine Readiness Assessment analyzes existing servers to check if they are compatible with the Azure platform. When you run the tool, you have to answer a series of questions about the workload on your virtual machine. Depending on the analysis of the running virtual machine and the responses to the questions, a detailed report is generated. The generated report contains suggestions for virtual machine configuration in Azure and hyperlinks to the articles that you can read before you migrate your workload.

Reference Link: https://azure.microsoft.com/downloads/vm-readiness-assessment/

You can use the Virtual Machine Readiness Assessment report to plan a migration to Azure:

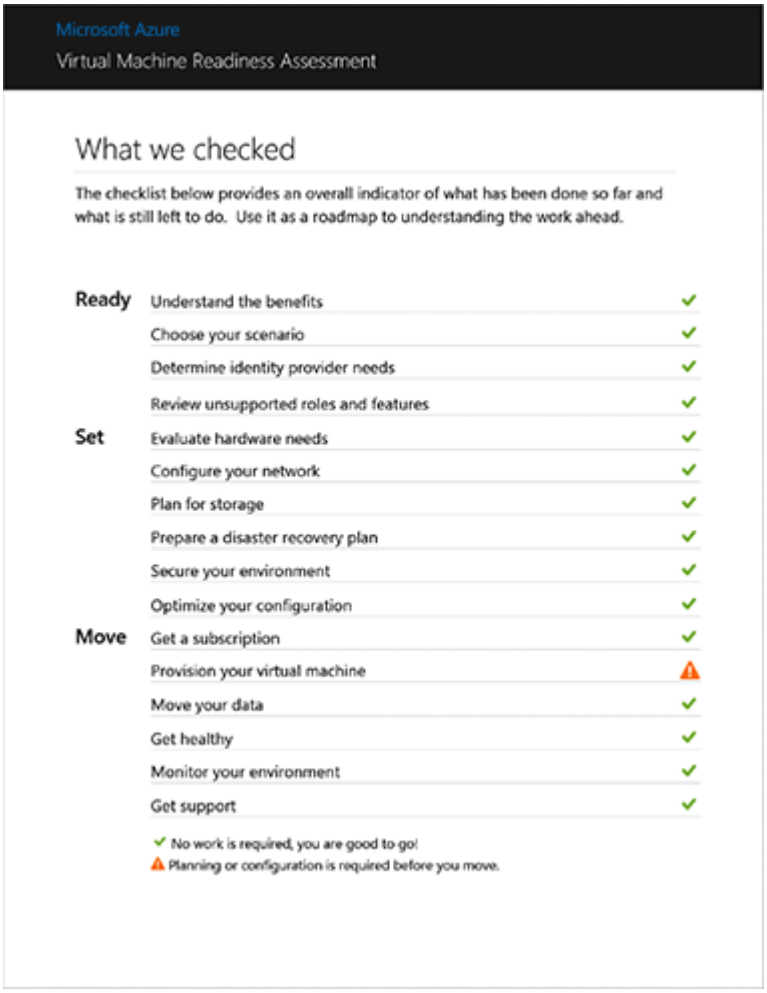


FIGURE 2.4: VIRTUAL MACHINE READINESS ASSESSMENT REPORT

Backup and Site Recovery

The Backup and Site Recovery services in Azure help you with simple back up and more advanced disaster recovery scenarios. These services are very flexible and you can use them on the on-premises servers or on the virtual machines in Azure.

Backup	Site Recovery
• Offsite backups of your server data	• Replicates private clouds to a secondary location
• Encrypted in transmission	• Quickly recover your virtual machines.
• Integrated in System Center and Windows Server	• Integrated with Windows Server Hyper-V Replica
• Backs up only changes instead of entire files	• Connects to System Center Virtual Machine Manager for health monitoring

Backup

Backup is a simple automated backup solution that uses other common Azure services and Windows Server features to minimize the amount of management that is required to back up data. Backup integrates directly with the data protection functionality of Windows Server and System Center. The backup data is stored in a storage

account that can be geo-replicated. Backup is optimized such that it only synchronizes incremental changes and all the data is secure during transit.

Reference Link: https://docs.microsoft.com/azure/backup/

Site Recovery

Site Recovery is a comprehensive disaster recovery service that provides deep orchestration and monitoring functionality. Site Recovery also uses existing technologies such as System Center, Hyper-V Replica, and SQL Server AlwaysOn. You can use recovery plans to orchestrate the recovery of your services by using distributed virtual machines and custom logic. You can regularly test recovery plans in isolation from your primary location.

Reference Link: https://docs.microsoft.com/azure/site-recovery/

Customizing Virtual Network Configuration by Using XML

When you create a virtual network, your services and virtual machines within the Virtual Network can communicate securely with each other without having to go out through the Internet. Because a cloud-only virtual network isn't intended for cross-premises connectivity, you won't need to acquire and configure a virtual private network (VPN) device or authentication certificates.

- Virtual Network settings can be represented as an XML file.
- This XML file (netcfg) can be used to initially configure a new virtual network
- A network configuration file (netcfg) can
- Network configuration files can be edited for an existing virtual network
- The netcfg schema is available on MSDN for reference

You can configure virtual networks in Azure during creation of the network by using either the Azure Management Portal or a network configuration file. The network configuration file is used by Azure to define the settings for your virtual network.

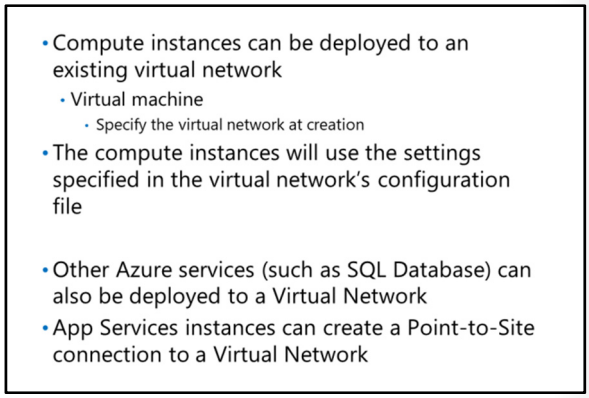
Network Configuration File

A network configuration file (netcfg) offers you the flexibility to customize your network settings and template networks. You can import a network configuration file into an existing virtual network to apply a specific configuration. You also can export a network configuration file from an existing virtual network to facilitate the creation of other virtual networks with similar settings. The actual file uses the .netcfg extension and is in the XML format. A schema for the network configuration file is available on MSDN. You can use this schema to create custom network configuration files by using a text editor.

Reference Link:
https://docs.microsoft.com/azure/virtual-network/virtual-networks-create-vnet-classic-netcfg-ps

Deploying Services to a Virtual Network

You can use virtual networks as the deployment targets for your virtual machines or database instance. When you create a virtual machine, you can specify a virtual network as the target to deploy the virtual machine's network interface card (NIC). You cannot virtual machines from one virtual network to another and you must recreate them in the new network. When you deploy virtual machines to a network, they receive the settings specified for that network. Other settings such as Network Security Groups are applied at the subnet level.



You can use the network configuration elements of the service configuration file to deploy cloud services into a virtual network that was previously defined in a network configuration file.

Some of the network configuration settings that you can specify in the service configuration file are:

- The DNS servers that can be used for name resolution
- The virtual network site name
- The address space and subnet for the cloud service

Lesson 4
Highly Available Azure Virtual Machines

You can use horizontal or vertical scaling with virtual machines and load balancing for a high availability scenario. Availability sets also affect the availability of your virtual machines during faults or upgrades.

This lesson describes the considerations for designing your virtual machine collections for high availability scenarios.

Lesson Objectives

After completing this lesson, you will be able to:

- Scale Virtual Machine instances horizontally or vertically.
- Describe how fault and upgrade domains affect virtual machine availability.

Availability Sets

Availability sets offer a mechanism that instructs Azure to place the virtual machines in separate fault or update domain. When a virtual machine stops working because of a fault (or for regular maintenance), at least one instance of your load balanced virtual machine remains available.

- Availability sets offers a mechanism to instruct Azure to place your virtual machines in separate fault or update domains
- When a virtual machine goes down because of a fault (or for regular maintenance), at least one instance of your load balanced virtual machines remains available
- Multiple virtual machine instances are required in order to be compatible with the Service Level Agreement (SLA)

Maintenance Events

In the Azure platform, planned maintenance can occur at any time. Typically, these updates are made by Microsoft to enable new features or improve reliability and performance. Also, the majority of these planned maintenance events are announced and do not have an impact on running virtual machines or cloud services. However, occasionally a planned maintenance event will require a reboot of a virtual machine or a restart of a cloud service.

Unplanned maintenance events can occur when there is a physical or hardware fault that affects your running Virtual Machine instance. Although the Azure platform automatically migrates your instances, there could potentially be downtime between the physical fault and the recovery orchestration.

Availability Sets

You can use availability sets to group two or more virtual machines in a manner that provides redundancy for an application. An availability set ensures that during planned or unplanned maintenance events, at least one virtual machine remains available. Typically, virtual machines that are part of the same tier of an application are placed into the same availability set.

Availability sets can be used to group virtual machines into application tiers:

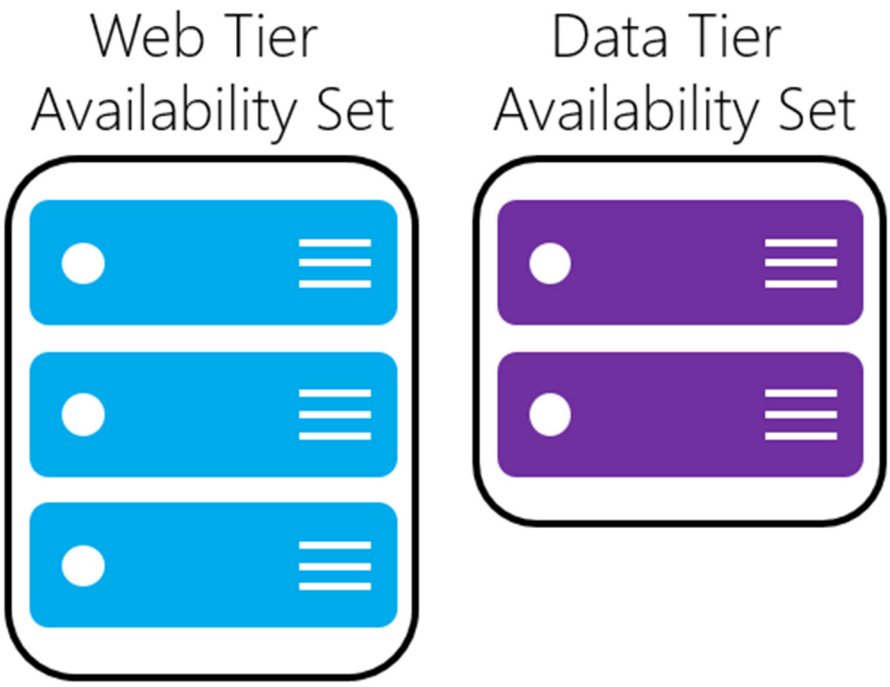


FIGURE 2.5: AVAILABILITY SETS EXAMPLE

Upgrade and Fault Domains

Virtual machines in your availability set are assigned an update domain and a fault domain by the Azure platform. An update domain is a logical unit that groups multiple virtual machines. Service instances within an update domain are typically upgraded together. This is especially common with the Cloud Services roles as they are updated and managed by the Azure platform. Fault domains refer to a physical point of failure. For example, a physical rack in a data center is a fault domain. It is ideal to group your services into multiple fault and update domains so that a small physical fault, such as hard drive failure, or an upgrade, will adversely impact all your application's instances at the same time.

In an availability set, five update domains are assigned by default. Six or more virtual machine instances are required before you start to see multiple instances in the same update domain. During an upgrade or reboot operation, only one update domain is rebooted at a time. This ensures consistent availability of your application throughout the operation. In the same availability set, two fault domains are assigned by default. This ensures that the virtual machine instances are in separate locations and that they are not impacted by the same physical fault, such as hardware failures, network outages, or a power interruption.

Fault domains and update domains in the Azure platform:

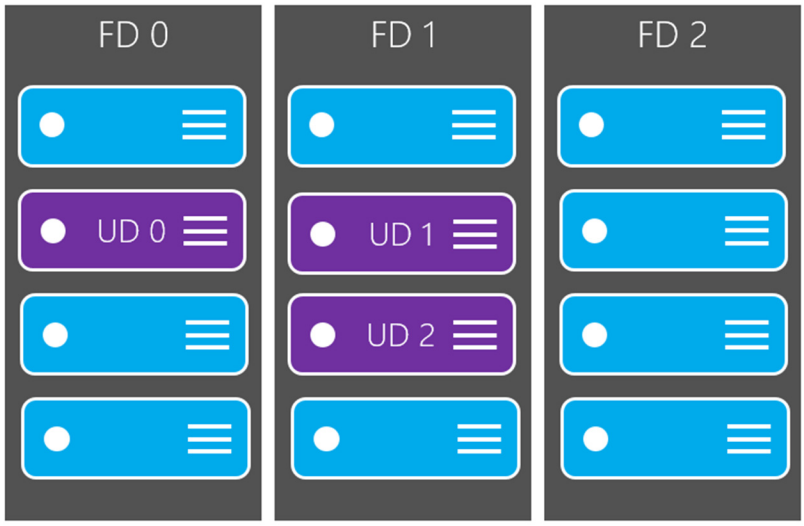


FIGURE 2.6: FAULT/UPDATE DOMAINS

Service Level Agreement

In an availability set, you must avoid leaving a single instance of a virtual machine by itself. A single virtual machine in an availability set does not qualify for a service level agreement (SLA) guarantee and will face down time during Azure's planned maintenance events. Your application may also face downtime when changing specific properties of the single virtual machine. Furthermore, if you deploy a single Virtual Machine instance within an availability set, you will not receive an advance warning or notification during platform maintenance. In this configuration, your single Virtual Machine instance can, and will, be rebooted with no advance warning when platform maintenance occurs.

Scaling Virtual Machines

Vertical Scaling

You can vertically scale virtual machine instances by changing their size. If there is an application that is running on a virtual machine and it requires more resources, you can handle it by increasing the virtual machine size. This can potentially add more memory, disks, or CPU cores.

- Virtual machines can be generally scaled in two directions
  - Horizontal
    - Duplicate virtual machine instances are added
  - Vertical
    - Virtual machine tiers can be changed for a current instance
- Virtual machines can be auto-scaled horizontally

Horizontal Scaling

If you have multiple Virtual Machine instances in an availability set, you can make use of the scaling functionality that is available in the Azure portals. You can manually change instance counts to scale an application up or down.

When you scale up or scale down horizontally an application that is running on virtual machines in an availability set, you can neither create new virtual machines nor delete the existing virtual machines. Instead you turn on or turn off any previously created virtual machines in an availability set. You can specify scaling based on the average percentage of CPU usage or based on the number of messages in a queue.

Manually changing the instance count:

FIGURE 2.7: INSTANCE COUNT

Autoscaling by Metric

On the Scale page, you can configure your cloud service to automatically increase or decrease the number of instances of virtual machines that are used by your application. You can configure scaling based on the following parameters:

- **Average CPU usage**. If the average percentage of CPU usage goes above or below specified thresholds, role instances are created or deleted, or virtual machines are turned on or turned off from an availability set.
- **Queue messages**. If the number of messages in a queue goes above or below a specified threshold, role instances are created or deleted, or virtual machines are turned on or turned off from an availability set.

When you define autoscale rules, you must specify the following values:

- **Schedule**. Defines the time and dates when the rules should be applied.
- **Instance Range**. The maximum and minimum number of instances to scale up or down to.
- **Metrics**
    - **Target CPU**. The ideal CPU range for the application's virtual machines. Scale up occurs when the average percentage of CPU usage is above the range and scale down occurs when the average percentage of CPU usage is below the range.
    - **Queue Target Per Machine**. The target number of queue messages for each virtual machine instance. Scale is determined by dividing the total number of queue messages by the ideal number of queue messages for each instance.
- **Scale Up/Down By**. The number of instances to start or stop for each scale operation.
- **Scale Up/Down Wait Time**. The minimum amount of time to wait before performing another scale operation.

Lesson 5
Virtual Machine Configuration Management

Although you can configure each virtual machine manually, automating the configuration of a virtual machine can lead to repeatable, efficient, and testable deployment scenarios. Automating configuration can also ensure that newly scaled instances of a virtual machine match the other instances.

This lesson discusses some of the most common methods of configuration management.

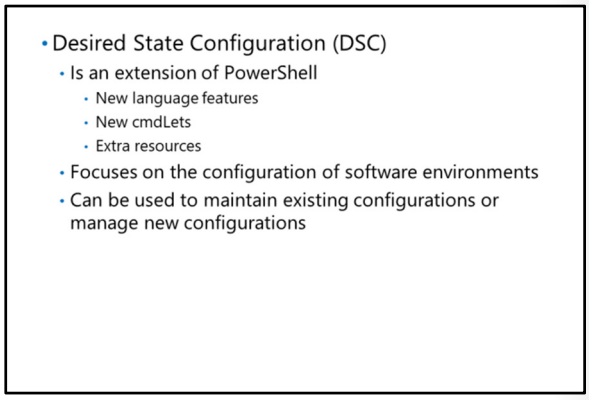
Lesson Objectives

After completing this lesson, you will be able to:

- Explain how Windows PowerShell desired state configuration (DSC) can be used for virtual machine configuration management.
- Describe the virtual machine agent service in Azure Virtual Machines.
- Describe the configuration management tools.

Windows PowerShell Desired State Configuration

Desired State Configuration (DSC) provides a set of Windows PowerShell language extensions, new Windows PowerShell cmdlets, and resources that you can use to declaratively specify how you want your software environment to be configured. It also provides a means to maintain and manage existing configurations.



DSC introduces a new keyword called **Configuration**. To use DSC to configure your environment, first define a Windows PowerShell script block by using the **Configuration** keyword, follow it with an identifier, and then with braces (**{}**) to delimit the block.

PowerShell DSC Configuration

```
Configuration WebServerConfig
{
   Node "WebServer"
   {
      WindowsFeature ServerRoleExample
      {
          Ensure = "Present"
          Name = "Web-Server"
      }
   }
}
```

You have now defined the example configuration. Next, you will need to enact this configuration. You can do this by invoking the configuration.

Invoking PowerShell DSC Configuration

```
PS C:\Scripts> WebServerConfig
```

Invoking the configuration creates Managed Object Format (MOF) files and places them in a new directory with the same name as the configuration block. The new MOF files contain the configuration information for the target nodes.

To enact the saved configuration, run the following command.

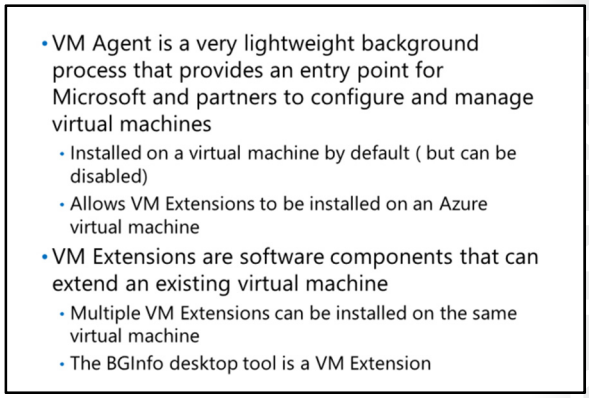
Enable Configuration

```
Start-DscConfiguration -Wait -Verbose -Path .\WebServerConfig
```

DSC includes other extensibility features such as parameters and nesting configurations.

VM Agent

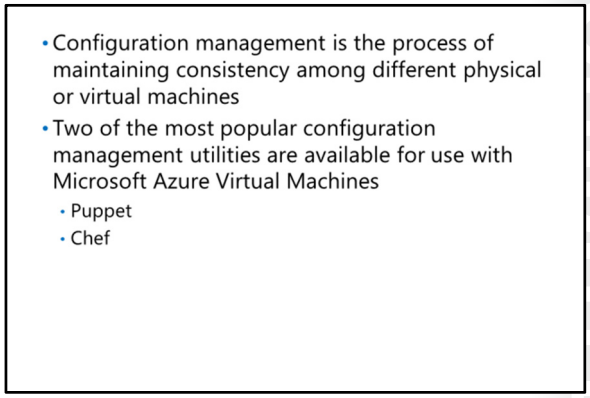
The virtual machine agent is a lightweight service that is typically installed on Azure virtual machines. This agent provides an extensibility point where extensions can be installed. Virtual machine extensions are custom extensions created that can be installed in Azure virtual machines, which have the virtual machine agent installed. By using virtual machine extensions, you can do the following:



- Automatically install custom or off-the-shelf software components in a virtual machine.
- Install, update, or remove custom features without having to recreate or update an existing virtual machine.
- Manage and view status or metrics for multiple virtual machines from a centralized tool or location.

Configuration Management Tools

Configuration management is the task of implementing, tracking, and controlling changes to software across a large variety of machines. Many of the common practices from source control management such as revision control are also seen in configuration management (CM) software. Configuration management utilities offers a lot of features such as:



- **Baselines**. Establishing a base configuration that is used as the default for new virtual or physical machines

- **Revision History**. Enables your team to determine who or what changes specific configuration settings
- **Replication**. Enables your configuration changes to be replicated across multiple machines
- **Agents**. Software specifically installed on machines to receive configuration change requests and apply them to the local machine

The configuration management utilities discussed in this topic support both Windows and Linux operating systems.

Chef and Puppet are two of the most common examples of configuration management software used throughout the industry. Both Chef and Puppet are written in Ruby and are licensed under the Apache license. Template images are available for both Chef and Puppet in Azure.

Puppet

Puppet is an open-source configuration management utility that is produced by Puppet Labs. Puppet has a unique declarative language that can be used to describe system configuration. These configuration changes can be applied directly to a virtual machine or distributed to multiple virtual machines by using a catalog. The Puppet agent periodically polls the machine for its current configuration and then syncs that configuration data to the Puppet master, a machine that manages all of the other machines with agents installed. The Puppet master ensures that the machines with the agent installed are in compliance with the latest configuration defined in the catalog.

Reference Link: http://puppetlabs.com/

Chef

Chef is another popular open-source configuration management utility. Chef is unique because configuration changes are composed into recipes. Recipes are composed of individual configuration changes that are called resources, which can include:

- A file to store
- A template configuration change
- A software package to install

Recipes can be combined and used to automate the most common infrastructure tasks along with software configuration changes. Using an agent called a node, the Chef server is polled for changes that are made to the installed recipes and ensures that individual machines (physical or virtual) are in compliance with the latest version of the recipe.

Reference Link: https://chef.io

Lesson 6
Customizing Azure Virtual Machine Networking

Although you can use Azure virtual machines right after you create them, you must perform additional configuration before interfacing these virtual machines instances with external resources or other virtual machines.

This lesson describes the methods used for customizing the network connectivity of an Azure virtual machine.

Lesson Objectives

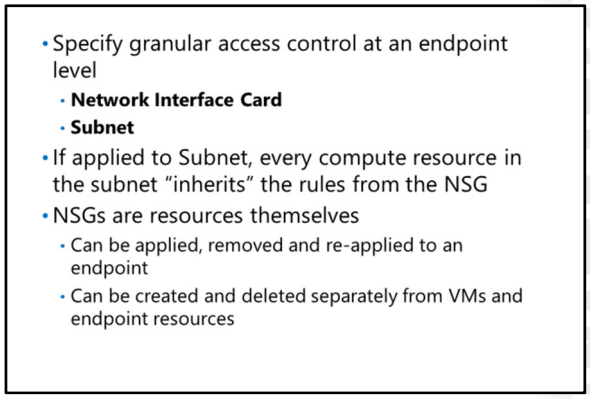
After completing this lesson, you will be able to:

- Use custom endpoints to expose public ports for a virtual machine.
- Customize the access control list for a virtual machine or cloud service.
- Modify Windows Firewall in a virtual machine.
- View the public Virtual IP Address (VIP) for a virtual machine.

Network Security Groups

An NSG works on one or more VM instances and controls all the traffic that is inbound and outbound.

You can associate an NSG to a VM, or to a subnet within a VNet. When associated with a VM, the NSG applies to all the traffic that is sent and received by the VM instance. When applied to a subnet within your VNet, it applies to all the traffic that is sent and received by ALL the VM instances in the subnet. A VM or subnet can be associated with only 1 NSG, and each NSG can contain up to 200 rules. You can have 100 NSGs per subscription on the VM.



A NSG is a top-level object that is associated to your subscription. An NSG contains access control rules that allow or deny traffic to VM instances. The rules of an NSG can be changed at any time, and changes are applied to all associated instances.

A network security group has a Name, is associated to a Region, and has a descriptive label. It contains two types of rules, Inbound and Outbound. The Inbound rules are applied on the incoming packets to a VM and the Outbound rules are applied to the outgoing packets from the VM. The rules are applied at the host where the VM is located. An incoming or outgoing packet has to match an Allow rule for it be permitted, if not it will be dropped.

Rules are processed in the order of priority. For example, a rule with a lower priority number (e.g. 100) is processed before rules with a higher priority numbers (e.g. 200). Once a match is found, no more rules are processed.

An NSG contains default rules. The default rules cannot be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create. The default rules describe the default settings recommended by the platform. While connectivity to the Internet is allowed for Outbound direction, it is by default blocked for Inbound direction. There is a default rule to allow Azure's load balancer (LB) to probe the health of the VM. You can override this rule if the VM or set of VMs under the NSG does not participate in the load balanced set.

Inbound

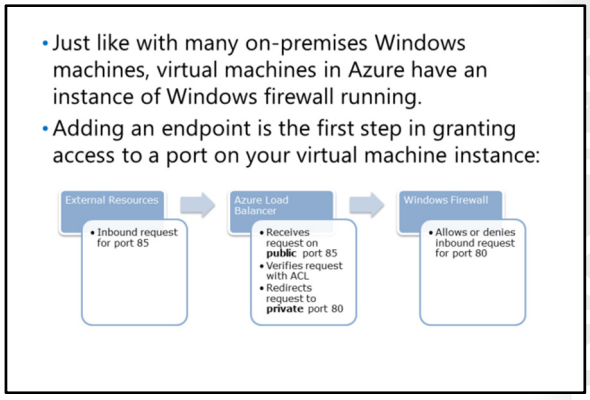
NAME	PRIORITY	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	PROTOCOL	ACCESS
ALLOW VNET INBOUND	65000	VIRTUAL_ NETWORK	*	VIRTUAL_ NETWORK	*	*	ALLOW
ALLOW AZURE LOAD BALANCER INBOUND	65001	AZURE_ LOADBALAN CER	*	*	*	*	ALLOW
DENY ALL INBOUND	65500	*	*	*	*	*	DENY

Outbound

NAME	PRIORITY	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	PROTOCOL	ACCESS
ALLOW VNET OUTBOUND	65000	VIRTUAL_ NETWORK	*	VIRTUAL_ NETWORK	*	*	ALLOW
ALLOW INTERNET OUTBOUND	65001	*	*	INTERNET	*	*	ALLOW
DENY ALL OUTBOUND	65500	*	*	*	*	*	DENY

Firewall Rules

Windows Firewall is a built-in, host-based, stateful firewall that is included in the client version of Windows and Windows Server. Windows Firewall drops incoming traffic that does not correspond to either the traffic that is sent in response to a request of the computer (solicited, outbound traffic) or unsolicited traffic that has been specified as allowed (excepted, inbound traffic). Windows Firewall is configured by using the Windows Firewall with Advanced Security snap-in, which integrates rules for both firewall behavior and traffic protection with Internet Protocol security (IPsec).

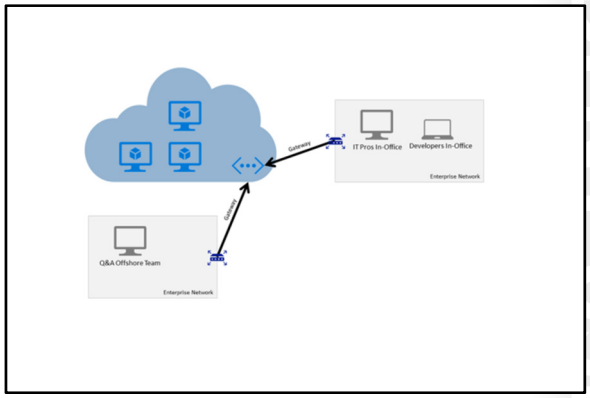


Virtual machines that are migrated to Azure should have their Windows Firewall rules verified prior to migration. It is possible that after a virtual machine is migrated to Azure, you might have issues connecting to the virtual machine for management tasks because of the Windows Firewall rules.

VNET Connectivity

Site-to-Site

A site-to-site VPN allows you to create a secure connection between your on-premises site and your virtual network. To create a site-to-site connection, a VPN device that is located on your on-premises network is configured to create a secure connection with the Azure Virtual Network Gateway. Once the connection is created, resources on your local network and resources located in your virtual network can communicate directly and securely. Site-to-site connections do not require you to establish a separate connection for each client computer on your local network to access resources in the virtual network.



Point-to-Site

A point-to-site VPN also allows you to create a secure connection to your virtual network. In a point-to-site configuration, the connection is configured individually on each client computer that you want to connect to the virtual network. Point-to-site connections do not require a VPN device. They work by using a VPN client that you install on each client computer. The VPN is established by manually starting the connection from the on-premises client computer. You can also configure the VPN client to automatically restart.

Note: Point-to-site and site-to-site configurations can exist concurrently.

Lesson 7
Virtual Machine Scale Sets

It is possible to create a large number of virtual machines and manage their installed software and configuration manually but it far from ideal. Virtual Machine Scale Sets is a service in Azure that can deploy large quantities of virtual machines for you, add them to a load balancer, configure them and auto-scale automatically for you.

This lesson will briefly introduce the Virtual Machine Scale Sets service and how you can create a VMSS instance for application workloads.

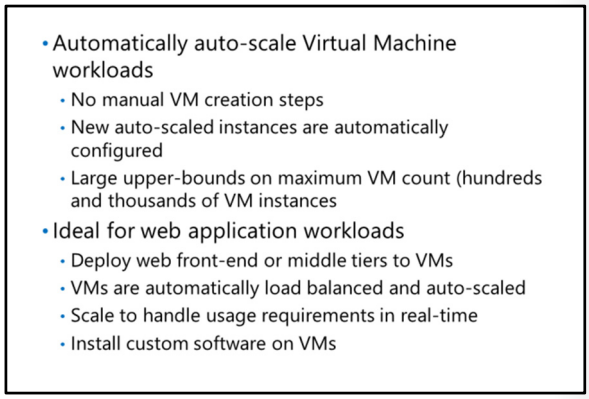
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the VMSS Service.
- Create a VMSS instance.

Virtual Machine Scale Sets (VMSS) Overview

An Azure VM Scale Set has several features that make it attractive to the Azure architect. The ability to define a VM Scale Set by JSON template and deploy it using any of the standard deployment methods enables their use in many automated solutions. This extends into continuous deployment scenarios with Visual Studio Team Services.



An Azure VM Scale Set allows a Virtual machine to deploy up to 1000 times in the same subnet in a controlled and automated manner with accurate auto-scaling.

An Azure VM Scale Set also requires no pre-provisioning of the Virtual Machine before adding to the scale set. The network and load balancer are created, configured and managed automatically, including the Network Address Translation (NAT) for access to and from the VM Instances.

These features added to the ease of deployment through the portal, Azure PowerShell or Azure CLI make the Azure VM Scale Set a powerful tool for the Azure cloud architect.

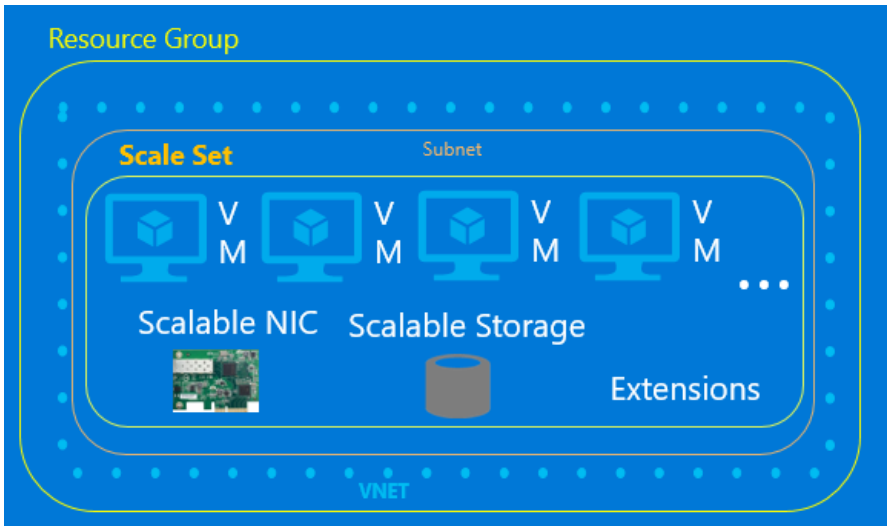


FIGURE 2.8: VIRTUAL MACHINE SCALE SET

Lab: Creating an Azure Virtual Machine for Development and Testing

Scenario

Before you begin the process of migrating your application from an on-premise server to Azure, you must create a development environment. You have elected to use Azure to host a Windows Server 2012 Virtual Machine. In this Virtual Machine, you will install project files, Visual Studio 2013 Update 4, Azure SDK for .NET 2.4 and Azure PowerShell. Once complete, you will use this virtual machine for all remaining development tasks.

Objectives

After you complete this lab, you will be able to:

- Create a virtual network.
- Create a Storage instance.
- Create a virtual machine.
- Manage the virtual machine VHDs.

Lab Setup

Estimated Time: 90 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Creating a Network and Resource Container

Exercise 2: Creating a Development Virtual Machine

Exercise 3: Configuring the Virtual Machine for Development

Module Review and Takeaways

In this module, you learned about IaaS offerings in Azure. The Virtual Network, Virtual Machines, Backup, and Site Recovery services provide several building blocks that you can use when you design a network of virtual machines or extend an existing datacenter.

Module 3

Hosting Web Applications on the Azure Platform

Contents:

Module Overview

This module provides an overview of the Azure Web Apps service. Lesson 1, "Azure Web Apps," introduces the Azure App Service platform-as-a-service offering available in Azure and specifically focuses on the Web Apps feature of App service. Lesson 2, "Azure Logic and Function Apps," explore two of the types of apps available in Azure App Service. Lesson 3, "Configuring an App Service App," discusses the various configuration options available to change the behavior of your app. Lesson 4, "Publishing an App Service App," describes the process for publishing a web application to an app. Lesson 5, "Supplemental Services," introduces additional service offerings for web applications in Azure such as the intelligent service offerings and the API Management service that can be used as a proxy to an App Service app.

Objectives

After completing this module, you will be able to:

- Create a Web App instance.
- Publish a simple ASP.NET web application to Web Apps.
- Monitor a Web App instance.
- Use additional Azure services with a Web App instance.
- Use Function and Logic Apps to create an integration workflow.

Lesson 1
Azure Web Apps

In many scenarios, it is preferable to use a quick and easy way to deploy web applications to the cloud rather than to reengineer the web applications as cloud projects. Web Apps allow you to quickly create a new Web App and iterate changes to the Web App in an agile manner.

This lesson describes the Web Apps service.

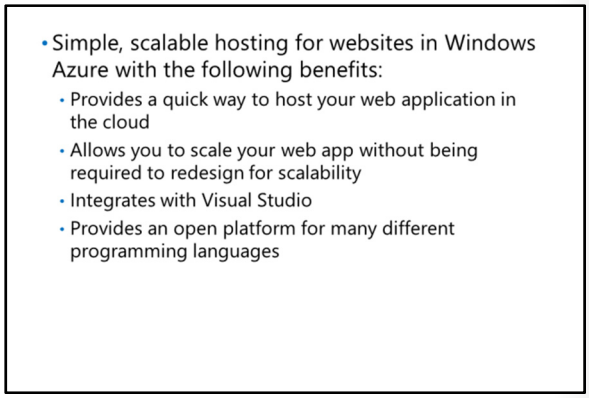
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Web Apps service.
- List the different tiers for a Web App.

Azure Web Apps Overview

Web Apps is a low friction Platform-as-a-Service (PaaS) offering to host your web applications in the Azure platform. The service is fully managed and you can easily configure advanced features such as AlwaysOn, custom domains, and auto-scale by using either portal.



Flexibility

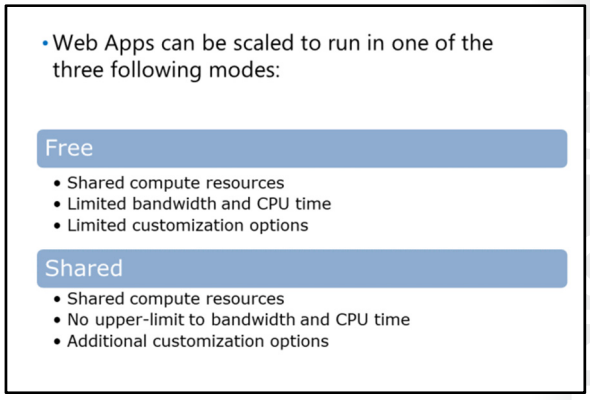
You can use a variety of integrated development environments (IDEs) and frameworks, such as .NET, Java, PHP, Node.js, or Python, to develop your web applications that are eventually deployed to Azure Web Apps. You can use Git and Kudu to deploy Node.js or PHP web applications. You also can deploy web applications that are developed in Microsoft Visual Studio to Web Apps by using the File Transfer Protocol (FTP) or the Web Deploy protocol.

Scalability

Because Web Apps is a fully managed service implementation, you can focus on developing your application and solving business problems instead of the hosting implementation and hardware scaling or specifics. You can easily scale up a stateless web application by configuring auto-scale in the portal. Auto-scale creates multiple instances of your Web App that are automatically load balanced so that your application can meet potential spikes in demand.

Web App Tiers

The Web Apps service is available in four tiers: Free, Shared, Basic, and Standard. You can use App Service Plans to assign a tier to a group of Web App instances. At any time, you can switch the tier for a App Service Plan. In the Free and Shared tiers, you are billed per hour for each instance of the Web App. In the Basic and Standard tiers, you are billed per hour for your dedicated virtual machine (compute instance) and not per Web App.



The Free tier offers 10 free Web App instances. All the instances share pool of60 minutes of CPU time per day. The free tier also enforces an outbound data limit of 165 megabytes (MB) per day. These instances are hosted on shared compute instances or virtual machines where they share resources with many other Web App tenants.

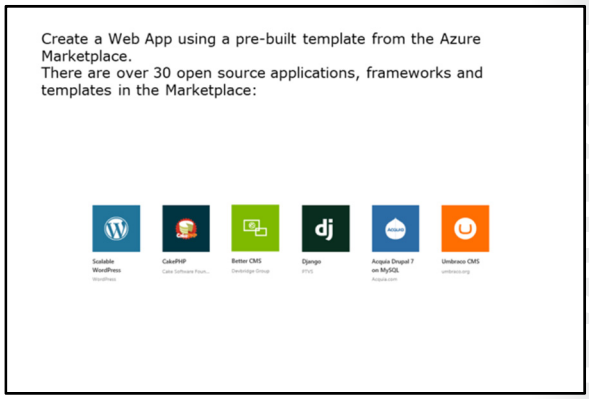
The Shared tier has a lot in common with the Free tier, but many restrictions are relaxed. For example, the Outbound Data limit is removed and each instance is allowed 240 minutes of CPU time per day that is not pooled. You can also have up to 100 Web App instances in this tier. In this tier, you also can use a custom domain with your Web App. You also can manually scale out to six different instances of your Web App. These Web App instances are hosted in different shared compute instances and are load balanced automatically.

In the Basic tier, your Web App instances are not in a shared environment. Instead, you have a dedicated compute instance where you can host as many Web App instances as you want. In addition to the features offered by the Free and Shared tiers, this tier also supports AlwaysOn, Secure Socket Layer (SSL) for custom domains, and a limited quantity of WebSocket connections (350 per Web App). You can manually scale out this tier to a maximum of three dedicated compute instances.

The Standard tier offers the same functionality as the Basic, Free and Shared tiers, but includes additional features such as Publishing Slots and Backups. With the auto-scale functionality, you can automatically scale your Web App based on a metric or a schedule.

Prebuilt Web App Templates

Most of the applications that are available in the Windows Web App Gallery () are also available in Microsoft Azure Marketplace.



Web App Configuration

Reference Link:
https://github.com/projectkudu/kudu/wiki

- The Web App deployment package and it's configuration are both stored in an external store.
- App Settings and Connection Strings are intercepted and changed in the application during startup
- Applications can be scaled by:
  - Creating IIS web sites using the Web Deploy package
  - Applying configuration options from the external store

Kudu is the engine behind most of the enhanced features that are offered by Web Apps. Kudu can also be used with your local web application projects. The Kudu project is an open-source project that is available on GitHub, and it supports many features such as:

- Web application publishing from a source control system
- Deployment hooks
- Web hooks
- Web Jobs

You can access the Kudu console for your Web App by using the following URL format:

https://[*Web App Name*].scm.azureWeb Apps.net

Kudu also provides an endpoint that the IIS Manager can use to remotely configure the web application. This provides a safer way to modify the application's Web.config file. Managing your web application using KUDU or the IIS Manager endpoint is not prone to errors unlike the traditional method of accessing the web application's file system and editing the configuration files.

Reference Link: https://azure.microsoft.com/blog/remote-administration-of-windows-azure-websites-using-iis-manager/

App Service Plans

- App Service Plans can logically group Web Apps within a subscription.
  - Characteristics such as features, capacity and tiers are shared amongst the Website instance in the group.
- Multiple App Service Plans can exist in a single Resource Group and multiple Web Appscan exist in a single App Service Plan.

You can group your Web App instances so that the capacity can be shared among them. At any given time, a Web App instance can only be associated with a single App Service Plan. A single Resource Group can contain multiple App Service Plans.

A App Service Plan is associated with a pricing tier:

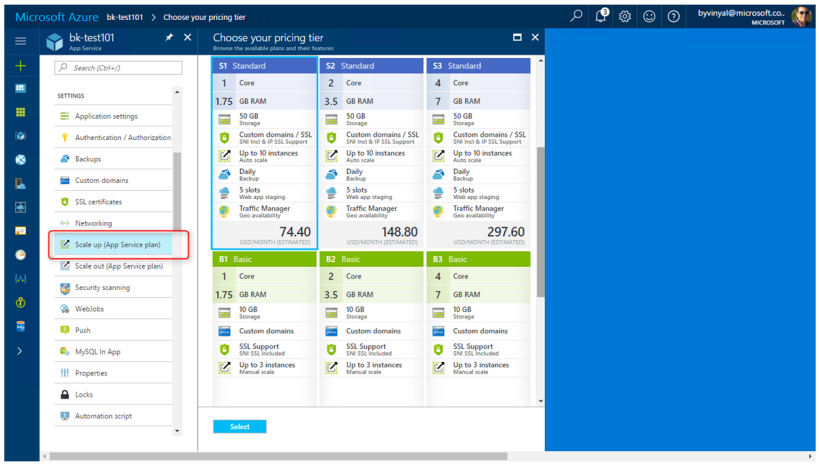


FIGURE 3.1: APP SERVICE PLAN PRICING TIERS

When you create a new Web App instance, you can either specify a App Service Plan or have a App Service Plan selected for you. By default, if your subscription does not have any App Service Plan, a new Standard tier App Service Plan is created automatically. If your subscription has an existing App Service Plan, then that plan is selected, by default, when you create a new Web App instance. You can also create a new App Service Plan when you create a new Web App instance.

All Web Apps within a App Service Plan are scaled together. Manual scale settings or auto-scale settings are configured on the entire App Service Plan. For example, you can configure a Standard App Service Plan to have a minimum of three instances and a maximum of five instances and auto-scale by monitoring the disk queue depth metric. Under an average load, you can expect to have four instances. This means that there are four dedicated compute instances and all your Web App instances have a copy on each dedicated instance that are load balanced by using the internal load balancer of Web Apps.

Reference Link: https://docs.microsoft.com/azure/monitoring-and-diagnostics/insights-how-to-scale

Lesson 2
Azure Logic and Function Apps

Serverless computing promises agility and power in building the next generation of solutions. You can use services such as Azure Functions or Azure Logic Apps to build these solutions. All of these services are useful when "gluing" together disparate systems. They can all define input, actions, conditions, and output. You can run each of them on a schedule or trigger. However, each service has unique advantages, and comparing them is not a question of "Which service is the best?" but one of "Which service is best suited for this situation?" Often, a combination of these services is the best way to rapidly build a scalable, full-featured integration solution.

This lesson describes the similarities and differences between the Azure Functions and Azure Logic Apps services.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure Functions service.
- Describe the Azure Logic Apps service.
- Compare the two services and decide when to use one or combine both in a solution.

Logic Apps

Logic Apps helps you build, schedule, and automate processes as workflows so you can integrate apps, data, systems, and services across enterprises or organizations. Logic Apps simplifies how you design and create scalable solutions for app integration, data integration, system integration, enterprise application integration (EAI), and business-to-business (B2B) communication, whether in the cloud, on premises, or both.

- No code designer for rapid creation
- Dozens of pre-built templates to get started
- Out of box support for popular SaaS and on-premises apps
- Use with custom API apps of your own
- Biztalk APIs for expert integration scenarios

For example, here are just a few workloads that you can automate with logic apps:

- Process and route orders across on-premises systems and cloud services.
- Move uploaded files from an FTP server to Azure Storage.
- Monitor tweets for a specific subject, analyze the sentiment, and create alerts or tasks for items that need review.

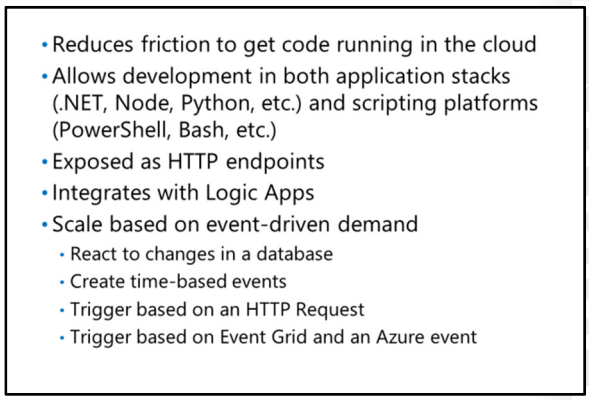
Every logic app workflow starts with a trigger, which fires when a specific event happens, or when new available data meets specific criteria. Many triggers include basic scheduling capabilities so that you can specify how regularly your workloads run. For more custom scheduling scenarios, start your workflows with the Schedule trigger.

Each time that the trigger fires, the Logic Apps engine creates a logic app instance that runs the workflow's actions. These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching. For example, this logic app starts with a Dynamics 365 trigger with the built-in criteria "When a record is updated". If the trigger detects an event that matches this criteria, the trigger fires and runs the workflow's actions. Here, these actions include XML transformation, data updates, decision branching, and email notifications.

You can build your logic apps visually with the Logic Apps Designer, available in the Azure portal through your browser and in Visual Studio. For more custom logic apps, you can create or edit logic app definitions in JavaScript Object Notation (JSON) by working in "code view" mode. You can also use Azure PowerShell commands and Azure Resource Manager templates for select tasks.

Function Apps

Azure Functions is a solution for easily running small pieces of code, or "functions," in the cloud. You can write just the code you need for the problem at hand, without worrying about a whole application or the infrastructure to run it. Functions can make development even more productive, and you can use your development language of choice, such as C#, F#, Node.js, Java, or PHP.



Azure Functions lets you develop serverless applications on Microsoft Azure.

Functions is a great solution for processing data, integrating systems, working with the internet-of-things (IoT), and building simple APIs and microservices. Consider Functions for tasks like image or order processing, file maintenance, or for any tasks that you want to run on a schedule.

Azure Functions integrates with various Azure and 3rd-party services. These services can trigger your function and start execution, or they can serve as input and output for your code. The following service integrations are supported by Azure Functions:

- Azure Cosmos DB
- Azure Event Hubs
- Azure Event Grid
- Azure Mobile Apps (tables)
- Azure Notification Hubs
- Azure Service Bus (queues and topics)
- Azure Storage (blob, queues, and tables)
- GitHub (webhooks)
- On-premises (using Service Bus)
- Twilio (SMS messages)

Azure Functions has two kinds of pricing plans. Choose the one that best fits your needs:

- **Consumption plan**: When your function runs, Azure provides all of the necessary computational resources. You don't have to worry about resource management, and you only pay for the time that your code runs.
- **App Service plan**: Run your functions just like your web, mobile, and API apps. When you are already using App Service for your other applications, you can run your functions on the same plan at no additional cost.

Lesson 3
Configuring an App Service App

App Service provides many features that you can use to expand your web application's capabilities. By using the Portal, you can enable different features of an App and modify the custom settings for the App without redeploying the web application.

This lesson lists the different configuration options that are available for an App.

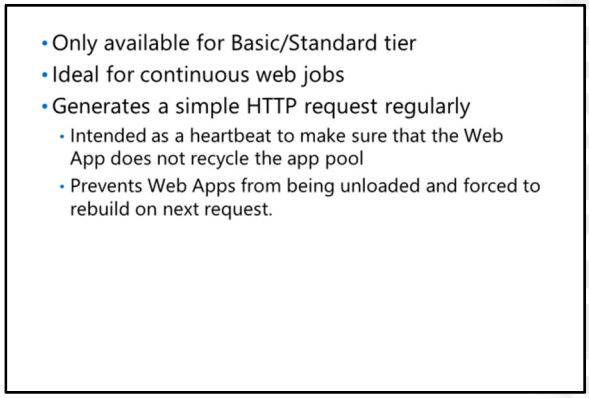
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the AlwaysOn feature.
- Describe how to use a custom domain with an App.
- Describe the auto-scale options for an App.

AlwaysOn

When you host your ASP.NET applications on IIS, you can isolate them into application pools. IIS improves the performance of your web server by automatically recycling application pools on a scheduled basis and by running the initial ASP.NET startup tasks for your web application only when the first request is processed. This prevents your application from using unnecessary resources when it is not requested by clients.



If your ASP.NET application is not precompiled, the application is compiled just-in-time (JIT) at startup and then the ASP.NET startup tasks will run. You can inject your own logic into ASP.NET's startup by implementing a Global.asax file with the **HttpApplication** class. If your ASP.NET application is precompiled, the startup tasks run immediately. Regardless of your choice, these startup tasks can be long and resource intensive. This will cause the first request or any request that is issued immediately after an application pool recycle to take a lot longer than normal to process.

Reference Link: https://docs.microsoft.com/aspnet/mvc/overview/getting-started/lifecycle-of-an-aspnet-mvc-5-application

In IIS, this can be resolved by setting your application pool's application start mode to Always Running. In Apps, you can accomplish this by using the AlwaysOn feature. AlwaysOn prevents your application from recycling due to going idle. AlwaysOn also improves your application startup time for early clients. This is accomplished by the Azure platform that regularly pings your App so that it is always active and is in a running state. This ensures that your application is already running before your first client requests are issued. It also ensures that your application remains in the running state and starts up in case of a recycle.

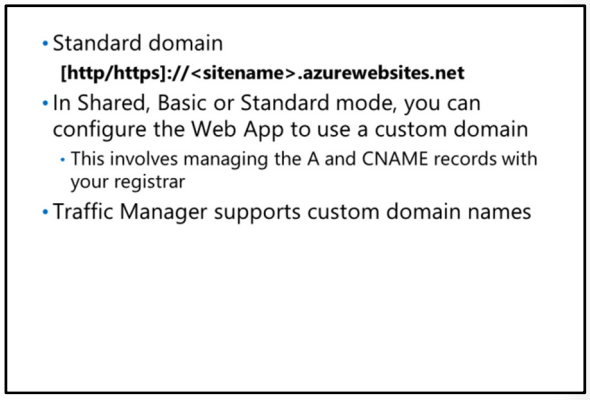
AlwaysOn is available only for the Basic and Standard tier Apps.

Domain Names

When you create a new App, a subdomain of the azurewebsites.net domain is assigned by using the following format:

[*http*|*https*]://<*sitename*>**.**azuresites.net

Azure also assigns a virtual Internet Protocol (IP) address for the same App instance. You can choose to use a custom domain name for your Web App and configure the custom domain name in the portal.



Note: Custom domain names are not supported for Free-tier App instances.

If you are using multiple instances within a single Web App, the virtual IP address will be load-balanced across those instances. You can use your domain registrar's Web App to configure a canonical name (CNAME) record and an address (A) records by using the following information.

HOST	RECORD TYPE	IP ADDRESS/URL
@	A	0.0.0.0 (IP address specified in portal)
www	CNAME	[*Web App name*].azureWeb Apps.net

After you complete this, you can use the same dialog box in the Azure Management portal to enable the custom domain on your Web App.

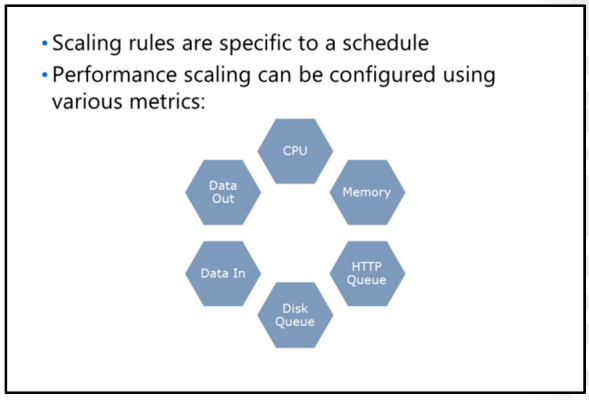
Reference Link:
https://docs.microsoft.com/azure/app-service/app-service-web-tutorial-custom-domain

If you want to host multiple App instances across regions, you can use a custom domain name with Microsoft Azure Traffic Manager.

Reference Link:
https://docs.microsoft.com/azure/app-service/web-sites-traffic-manager-custom-domain-name

Autoscaling Apps

In many distributed application scenarios, you might want to scale out (horizontally) your application by increasing the number of instances of the application. Using the built-in load balancer, you can spread the load for your web application across multiple instances. This minimizes the per instance cost and ensures that your application meets the increasing demand from client devices or browsers.



With unpredictable application workloads, you can sometimes end up overestimating or underestimating the number of Apps that are needed to provide the best user experience. Traditionally, overestimation is used to ensure that each user has a satisfactory experience with your web application. Ideally, you want your web platform to use the extra instances only when it is necessary and to shut down the same instances when they are no longer needed.

Common web application computing patterns:

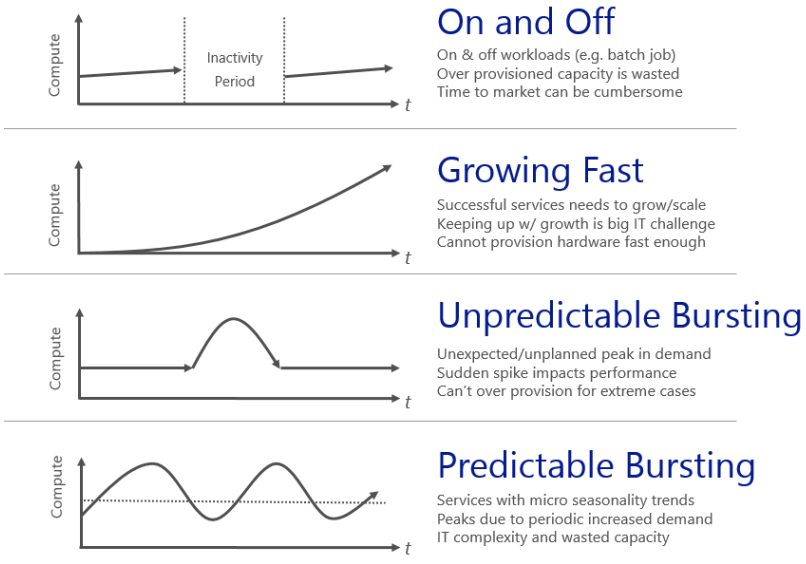


FIGURE 3.2: COMPUTING PATTERNS

You can use auto-scale to control horizontal scaling by using metrics and schedules. This gives you the flexibility to have your application's resource allocation closely aligned with the actual utilization. By using auto-scale, you can:

- Minimize unnecessary resource cost by removing App instances when they are no longer needed.
- Maximize performance and client response by creating App instances when a measured threshold is met.

When you define the auto-scale configuration, you must specify the schedules. You can use schedules to specify different auto-scale rules for different date and time periods. By default, a schedule is created for *all time*. After you create or select a schedule, you can define a metric to measure configuration values such as:

- CPU Utilization Percentage (range)
- Storage Queue Length (threshold)

After you save this configuration, you can monitor your auto-scale history for your Web App by using the same configuration dialogs. In the Portal, an enhanced list of auto-scale metrics is available that you can use to scale your App such as:

- Average Memory
- HTTP Queue Depth
- Disk Queue Depth

You also can use the Portal to specify more options that determine how long the web application has to wait before scaling up or how long it has to wait between scale actions.

 **Reference Link:** https://docs.microsoft.com/azure/monitoring-and-diagnostics/insights-how-to-scale

Lesson 4
Publishing an Azure App Service App

After developing a web application, you can use Web Deploy to publish the application to Azure. By using the publish wizard in Visual Studio, you can customize the configuration settings and connection strings before you publish the web application.

This lesson will focus on the deployment options for App Service Apps in Azure.

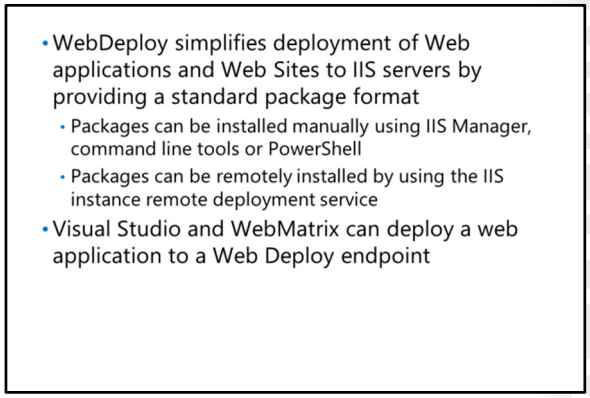
Lesson Objectives

After completing this lesson, you will be able to:

- Describe how to publish a web application by using the publish wizard in Visual Studio.
- Describe how to transform the configuration settings of your web application.
- Describe the difference between the standard Release and Debug builds.

The Web Deploy Protocol

Web Deploy (msdeploy) is a combination of a package format and an IIS add-in that gives administrators and developers incredible flexibility to manage and deploy container applications.



A Web Deploy package is a simple representation of an IIS Web Application. The package can contain the following data about your Web application:

- Binaries
- Content
- XML configuration
- Databases
- Registry modifications
- Assemblies and Globally Assembly Cache (GAC) references

You can create Web Deploy packages manually or by using an IDE such as Visual Studio or WebMatrix. You can hand off the package to an administrator asynchronously so that he or she can install the package in any IIS install with the Web Deploy add-in. At installation, the administrator provides configuration values for items such as SQL database connection strings. You also can use Web Deploy can to synchronize changes to an application among a server farm. You can extract a Web Deploy package from an existing IIS Web App and import it into IIS Web Applications that are on other machines within the same farm. Web Deploy can also expose an endpoint that allows developers to remotely deploy applications to a web server without having direct access to the actual web server.

You can use Web Deploy packages to synchronize development, test, staging, and production environments regardless of where they are hosted:

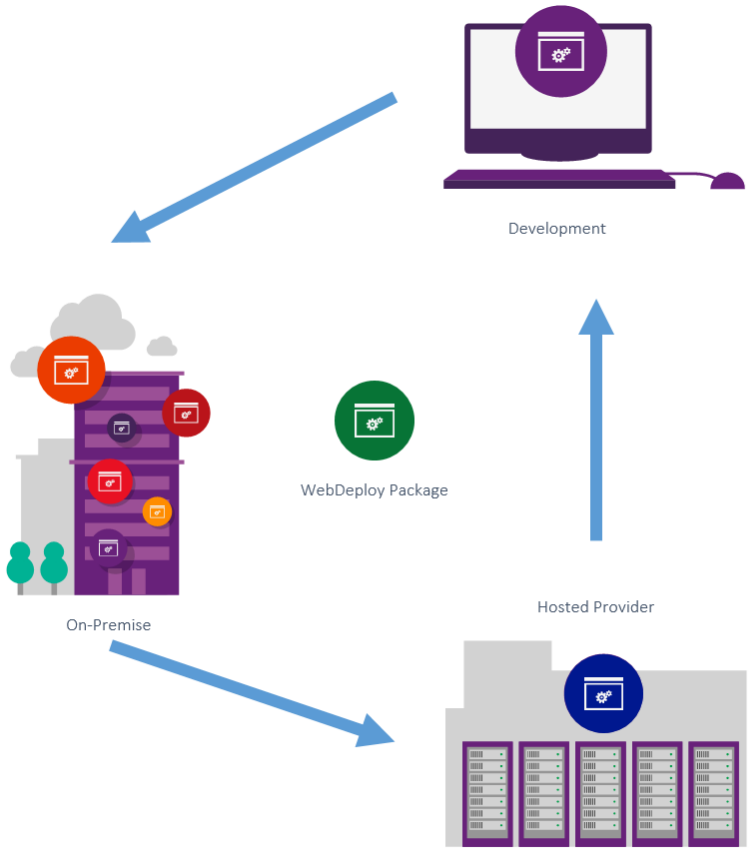


FIGURE 3.3: WEB DEPLOY ECOSYSTEM

Azure App Service allows developers to publish a Web Deploy package to an App instance by using the remote deploy service. Connection strings and databases are already managed for you to provide a seamless publish experience.

Reference Link: https://docs.microsoft.com/azure/app-service/app-service-deploy-local-git#ide

Reference Link: https://docs.microsoft.com/iis/install/installing-publishing-technologies/installing-and-configuring-web-deploy-on-iis-80-or-later

Lesson 5
Supplemental Services

Many applications can benefit from the inclusion of first-party and third-party services available in the cloud. Whether the services add intelligent features to an application, or provides additional security and management features, these services can be interwoven into a variety of application scenarios.

This lesson focuses on various intelligent services such as Cognitive Services, Bot Service, LUIS and QnA Maker and the API Management proxy service.

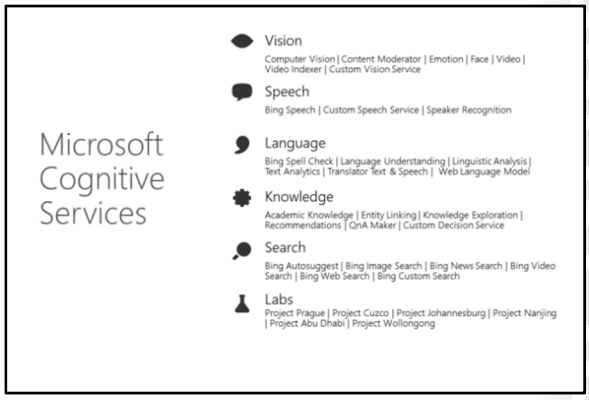
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the APIs available in Azure Cognitive Services and the related Bing APIs.
- Describe the relationship between Bot Service and Bot Framework.
- Use Azure API Management to extend, enhance and protect and existing API endpoint.

Cognitive Services

Microsoft Cognitive Services are a set of APIs, SDKs and services available to developers to make their applications more intelligent, engaging and discoverable. Microsoft Cognitive Services expands on Microsoft's evolving portfolio of machine learning APIs and enables developers to easily add intelligent features—such as emotion and video detection; facial, speech and vision recognition; and speech and language understanding—into their applications.



Bing APIs

Cognitive Services, as a suite, also includes various Bing APIs that can be used in your applications:

- **Bing Web Search**
  - Bing Web Search API provides an experience similar to Bing.com/search by returning search results that Bing determines are relevant to a user's query. The results include Web pages and may also include images, videos, and more.
- **Bing Image Search**
  - Bing Image Search API provides an experience similar to Bing.com/images by returning images that Bing determines are relevant to a user's query.
- **Bing Autosuggest**
  - Bing Autosuggest API lets you send a partial search query term to Bing and get back a list of suggested queries that other users have searched on. For example, as the user enters each character of their search term, you'd call this API and populate the search box's drop-down list with the suggested query strings.

QnA Maker

Microsoft QnA Maker is a REST API and web-based service that trains AI to respond to user's questions in a more natural, conversational way. QnA Maker provides a graphical user interface that allows non-developers to train, manage, and use the service for a wide range of solutions.

QnA Maker extracts a knowledge base from two types of input: FAQ pages and product manuals. The tool supports extraction from FAQ web pages or documents in the question-answer format. The tool can also extra QnA pairs from PDF-format product manuals.

Once extracted, the QnA Maker service creates a knowledge base and bot using the knowledge base. The bot can then be used, via a REST API, in any existing web application or website to answer questions for users. Over time, the knowledge base can be updated, retrained, and republished to meet the morphing needs to a user-facing web application.

Language Understanding (LUIS)

Language Understanding (LUIS) allows your application to understand what a person wants in their own words. LUIS uses machine learning to allow developers to build applications that can receive user input in natural language and extract meaning from it. A client application that converses with the user can pass user input to a LUIS app and receive relevant, detailed information back.

A LUIS app is a domain-specific language model designed by you and tailored to your needs. You can start with a prebuilt domain model, build your own, or blend pieces of a prebuilt domain with your own custom information.

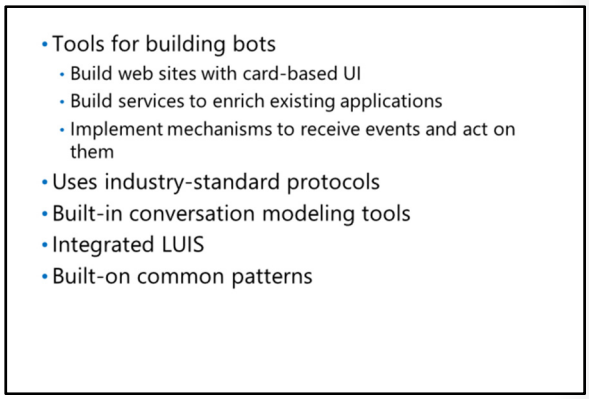
A model starts with a list of general user intentions such as "Book Flight" or "Contact Help Desk." Once the intentions are identified, you supply example phrases called utterances for the intents. Then you label the utterances with any specific details you want LUIS to pull out of the utterance.

Prebuilt domain models include all these pieces for you and are a great way to start using LUIS quickly.

After the model is designed, trained, and published, it is ready to receive and process utterances. The LUIS app receives the utterance as an HTTP request and responds with extracted user intentions. Your client application sends the utterance and receives LUIS's evaluation as a JSON object. Your client app can then take appropriate action.

Bot Services

Bot Service provides an integrated environment that is purpose-built for bot development, enabling you to build, connect, test, deploy, and manage intelligent bots, all from one place. Bot Service leverages the Bot Builder SDK with support for .NET and Node.js. You can write a bot, connect, test, deploy, and manage it from your web browser with no separate editor or source control required. For simple bots, you may not need to write code at all. Bot Service accelerates bot development with Five bot templates you can choose from when you create a bot. You can further modify your bot directly in the browser using the Azure editor or in an Integrated Development Environment (IDE), such as Visual Studio and Visual Studio Code.

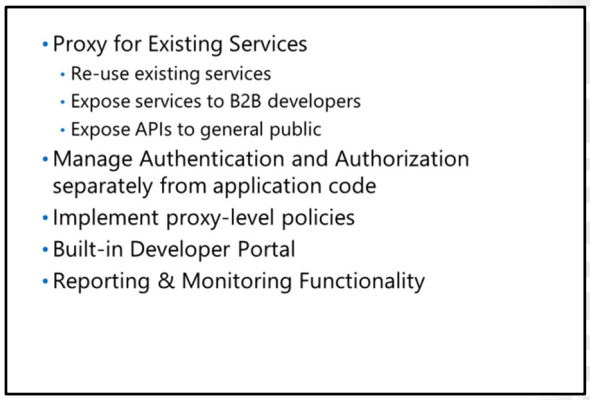


Here are some key features of Bot Service:

- **Multiple language support**
  - Bot Service leverages Bot Builder with support for .NET and Node.js.
- **Bot templates**
  - Bot Service templates allow you to quickly create a bot with the code and features you need. Choose from a Basic bot, a Forms bot for collecting user input, a Language understanding bot that leverages LUIS to understand user intent, a QnA bot to handle FAQs, or a Proactive bot that alerts users of events.
- **Bring your own dependencies**
  - Bots support NuGet and NPM, so you can use your favorite packages in your bot.
- **Flexible development**
  - Code your bot right in the Azure portal or set up continuous integration and deploy your bot through GitHub, Visual Studio Team Services, and other supported development tools. You can also publish from Visual Studio.
- **Connect to channels**
  - Bot Service supports several popular channels for connecting your bots and the people that use them. Users can start conversations with your bot on any channel that you've configured your bot to work with, including Skype, Facebook, Teams, Slack, SMS, and several others.
- **Tools and services**
  - Test your bot with the Bot Framework Emulator and preview your bot on different channels with the Channel Inspector.
- **Open source**
  - The Bot Builder SDK is open-source and available on GitHub.

API Management

API Management (APIM) helps organizations publish APIs to external, partner, and internal developers to unlock the potential of their data and services. Businesses everywhere are looking to extend their operations as a digital platform, creating new channels, finding new customers and driving deeper engagement with existing ones. API Management provides the core competencies to ensure a successful API program through developer engagement, business insights, analytics, security, and protection. You can use Azure API Management to take any backend and launch a full-fledged API program based on it.



To use API Management, administrators create APIs. Each API consists of one or more operations, and each API can be added to one or more products. To use an API, developers subscribe to a product that contains that API, and then they can call the API's operation, subject to any usage policies that may be in effect. Common scenarios include:

- **Securing mobile infrastructure**: by gating access with API keys, preventing DOS attacks by using throttling, or using advanced security policies like JWT token validation.
- **Enabling ISV partner ecosystems**: by offering fast partner onboarding through the developer portal and building an API facade to decouple from internal implementations that are not ripe for partner consumption.
- **Running an internal API program**: by offering a centralized location for the organization to communicate about the availability and latest changes to APIs, gating access based on organizational accounts, all based on a secured channel between the API gateway and the backend.

The system is made up of the following components:

- The **API gateway** is the endpoint that:
  - Accepts API calls and routes them to your backends.
  - Verifies API keys, JWT tokens, certificates, and other credentials.
  - Enforces usage quotas and rate limits.
  - Transforms your API on the fly without code modifications.
  - Caches backend responses where set up.
  - Logs call metadata for analytics purposes.
- The **Azure portal** is the administrative interface where you set up your API program. Use it to:
  - Define or import API schema.
  - Package APIs into products.
  - Set up policies like quotas or transformations on the APIs.
  - Get insights from analytics.
  - Manage users.
- The **Developer portal** serves as the main web presence for developers, where they can:
  - Read API documentation.
  - Try out an API via the interactive console.
  - Create an account and subscribe to get API keys.
  - Access analytics on their own usage.

Developer Portal

The developer portal is where developers can learn about your APIs, view and call operations, and subscribe to products. Prospective customers can visit the developer portal, view APIs and operations, and sign up. The URL for your developer portal is located on the dashboard in the Azure portal for your API Management service instance.

Products

Products are how APIs are surfaced to developers. Products in API Management have one or more APIs, and are configured with a title, description, and terms of use. Products can be Open or Protected. Protected products must be subscribed to before they can be used, while open products can be used without a subscription. When a product is ready for use by developers, it can be published. Once it is published, it can be viewed (and in the case of protected products subscribed to) by developers. Subscription approval is configured at the product level and can either require administrator approval, or be auto-approved.

Policies

Policies are a powerful capability of API Management that allow the Azure portal to change the behavior of the API through configuration. Policies are a collection of statements that are executed sequentially on the request or response of an API. Popular statements include format conversion from XML to JSON and call rate limiting to restrict the number of incoming calls from a developer, and many other policies are available.

Lesson 6
Lab Overview

This lesson provides a high-level overview of the Contoso Events web application. In this lesson, you will see a demonstration of the Administration web application and the public-facing web front end of the Contoso Events web application.

Lesson Objectives

After completing this lesson, you will have an understanding of how the Contoso.Events web solution works.

Lab: Creating an ASP.NET Web App by Using Azure Web Apps

Scenario

You have an events administration application that is currently used by a static set of users. The application must be upgraded to handle all the users in your organization in the future. You need a hosting option that provides the least amount of friction so that you can immediately deploy the web application for immediate use. You also need the hosting option to be flexible enough so that it allows you to configure and scale the web application, thereby ensuring that it can handle an increase in the number of administrative users. For these reasons, you have chosen to deploy the application to Web Apps. Web Apps will also give you the flexibility to integrate your application with Azure Active Directory in the future so that all of your organization's users can access the application.

In this lab, you will create a Web App, deploy your existing application, and then configure the Web App after deployment.

Objectives

After you complete this lab, you will be able to:

- Create a Web App.
- Create a linked resource for a Web App.
- Publish an ASP.NET web application to a Web App.
- Modify the configuration of the Web App in the Management Portal.

Lab Setup

Estimated Time: 60 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Creating an Azure Web App and Function App

Exercise 2: Deploying an ASP.NET Web Application to an Azure Web App

Exercise 3: Configuring an Azure Web App

Exercise 4: Deploying a Console Application to an Azure Function App

Verify the correctness of the statement by placing a mark in the column to the right.

Statement	Answer
True or False: WebDeploy is the only mechanism for deploying web applications to Web Apps.	

Module Review and Takeaways

In this module, you learned about Web Apps and how easy it is to migrate an ASP.NET web application project to the cloud. You also learned about the unique configuration options and features that make Web Apps a really powerful way of hosting cloud-based web applications. Finally, you learned about the monitoring tools, available out of the box with Web Apps, that you can use to debug and trace Web Apps in Azure.

Best Practice

To deploy applications in an on-premises environment, you can follow the same process that is used to publish a web application to Azure by using Web Deploy. Web Deploy can be used to decouple developers from the release process. It provides a single package that can be used by an administrator to deploy the application in the production environment.

Review Questions

Question: What are some of the business scenarios where you can scale your Web App on a schedule?

Question: Why would you want to store logs in a central location for multiple Web App instances?

Question: Why would you consider leaving log files on the file system for a Web App instance?

Module 4

Storing SQL Data in Azure

Contents:

Module Overview

Dynamic web applications must store the data that is being managed and manipulated by end users. ASP.NET technologies such as ADO.NET and Entity Framework provide a way for accessing data in SQL Server. In the cloud, the Microsoft Azure platform provides a database as a service offering that allows developers to use SQL in the same way as they would in an on-premises location. Lesson 1, "Azure SQL Database Overview," describes the Azure SQL Database service and reasons you would consider using it. Lesson 2, "Managing SQL Databases in Azure," describes the familiar and new management tools that are available for use with a SQL database that is hosted in Azure. Lesson 3, "Azure SQL Database Tools," describes the SQL Server Data Tools (SSDT) templates, panes, and projects that are available in Microsoft Visual Studio 2013. Lesson 4, "Securing and Recovering an Azure SQL Database Instance," describes the recovery scenarios relevant in Azure SQL Database. Lesson 5, "Azure Database for MySQL and PostgreSQL," introduces the two managed database options for PostgreSQL and MySQL hosting.

Objectives

After completing this module, you will be able to:

- Describe the difference between Azure SQL Database editions.
- Explain some of the advantages and disadvantages of hosting databases in Azure SQL Database.
- Explain some of the advantages and disadvantages of hosting databases in a SQL Server installation on a virtual machine in Azure.
- Describe the tools that you can use to manage Azure SQL Database.
- Implement a high-availability solution with Azure SQL Database.
- Describe the Azure Database for MySQL and PostgreSQL services.

Lesson 1
Azure SQL Database Overview

SQL Database provides a database-as-a-service offering that allows you to take advantage of many features that are familiar to developers and administrators who worked with SQL Server. SQL Database exposes a tabular data stream (TDS) endpoint so that you can use many of your existing tools to connect to and manage your SQL Database instance.

This lesson describes the SQL Database service, some of the advantages of using this service, and some of the considerations for selecting between the SQL Database service and SQL Server on a virtual machine in Azure.

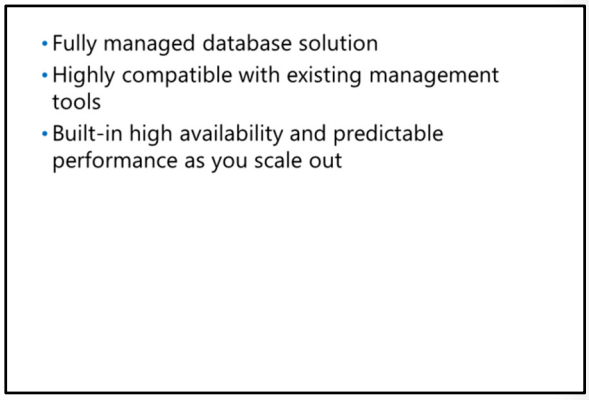
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the SQL Database service.
- Explain the advantages of using SQL Database.
- Explain the advantages of using SQL Server that is installed on a Virtual Machine in Azure.
- Compare the service tiers.
- Describe the SQL Database editions.

Azure SQL Database

SQL Database is a relational database as a service offering that provides predictable performance and a high degree of compatibility with existing management tools.



Predictable Performance

By using a consistent unit of measurement, such as Database Throughput Units, you can compare the expected service level for each performance tier that is offered in the SQL Database service. Consistent and predictable performance allows you to select a tier that very closely matches your application's real-world utilization.

High Compatibility

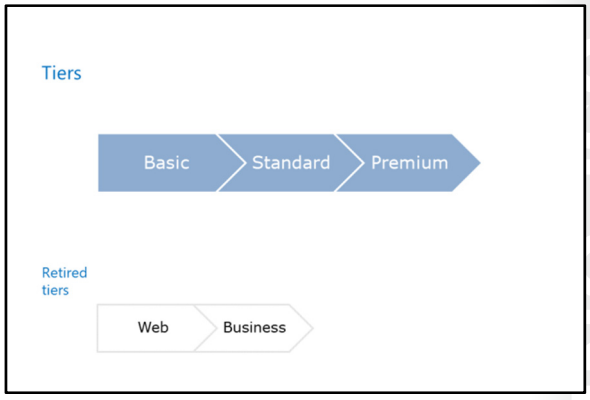
A Tabular Data Stream (TDS) endpoint is provided for each logical server that is created in the SQL Database service. You can use existing SQL client applications and tools with SQL Database by using the TDS protocol.

Simple Management

Additional tools are available in Azure to manage databases that are created by SQL Database. A portal for managing database objects is available in the Azure Management Portal, which you can access by clicking the Manage button. You also can manage SQL Database instances by using the portals, REST API, Windows PowerShell, or the cross-platform command-line interface (Xplat CLI).

Azure SQL Database Tiers

The SQL Database service is offered in several tiers. You can select a tier that closely matches your application's intended or actual resource needs. The following is a list of SQL Database service tiers with the associated performance characteristics:



- **Basic**: Ideal for simple databases that requires only a single connection performing a single operation at a time.
- **Standard**: The most common option and is used for databases that require multiple concurrent connections and operations.
- **Premium**: Designed for applications that require large quantities of transactions at volume. These databases support a large quantity of concurrent connections and parallel operations.

These tiers are further separated into performance levels. Performance levels are very specific categories within a service tier that provides a specific level of service. For example, the P1 performance level in the Premium tier offers a maximum database size of 500 gigabyte (GB) and a benchmarked transaction rate of 105 transactions per second.

Database Throughput Unit (DTU)

DTUs are used to describe the capacity for a specific tier and performance level. DTUs are designed to be relative so that you can directly compare the tiers and performance levels. For example, the Basic tier has a single performance level (B) that is rated at 5 DTU. The S2 performance level in the Standard tier is rated at 50 DTU. This means that you can expect ten times the power for a database at the S2 performance level than a database at the B performance level in the Basic tier.

Tiers and Performance Levels

Every tier has one or more performance levels. In general, the performance levels in the Premium tier are rated higher than the performance levels in the Standard tier, which are again rated higher than the Basic tier. The following chart illustrates this distinction.

Reference Link: https://docs.microsoft.com/azure/sql-database/sql-database-service-tiers

	DATABASE THROUGHPUT UNITS	DATABASE SIZE	POINT IN TIME RESTORE
B	5	2 GB	7 Days
S0	10	250 GB	14 Days
S1	20	250 GB	14 Days
S2	50	250 GB	14 Days
S3	100	250 GB	14 Days
P1	100	500 GB	35 Days

	DATABASE THROUGHPUT UNITS	DATABASE SIZE	POINT IN TIME RESTORE
P2	200	500 GB	35 Days
P3	800	500 GB	35 Days

Databases as a Service vs. SQL Server in a Virtual Machine

SQL Database will always be compared with SQL Server Standalone that is hosted on a physical or virtual machine. With the SQL Database service, the majority of administration tasks are managed entirely by Microsoft. In scenarios where you require the ability to perform very unique and custom configuration, analysis, or administration, you can always host the SQL Server standalone product in a virtual machine in the Azure IaaS platform.

SQL Database	SQL Server in an Azure VM
• Standardized, interoperable, and scalable managed database solution • Contains size limits for each standard edition • Requires some re-architecture of existing applications • Ideal for new cloud-based applications	• Provides high compatibility with SQL on premise. • Ideal for existing applications that require the SQL installation to be customized. • Requires more maintenance and customization to achieve scalability

SQL hosted in the Microsoft data platform:

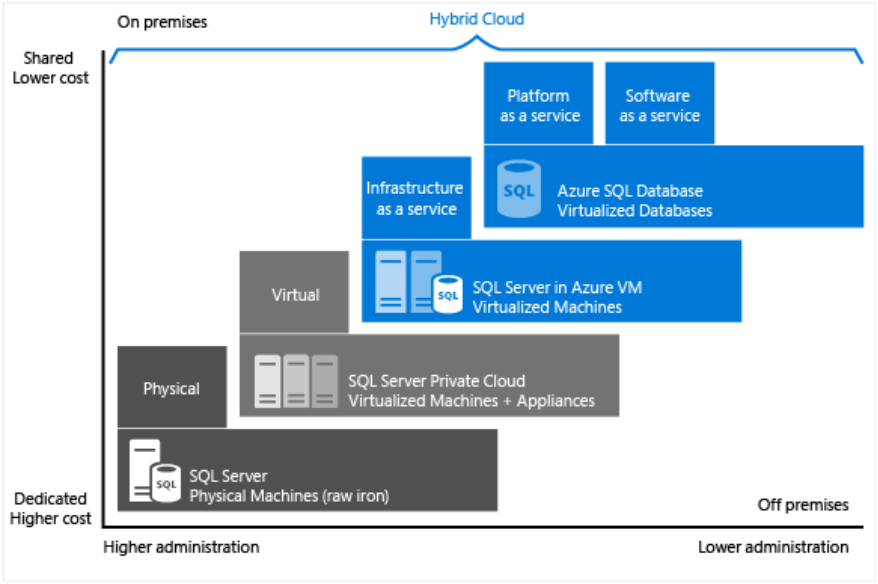


FIGURE 4.1: SQL DATA PLATFORM

Hosting Data in SQL Database

- This solution is ideal for new applications that are based on cloud technologies, where requirements dictate that you do not need to customize the behavior of SQL Server. Many applications use SQL Server for storage, programmatic functionality, and basic query operations.
- The databases that are stored in SQL Database are constrained by size. Solutions such as Elastic Scale are available to scale databases across multiple instances of different tiers.

Hosting Data in SQL Server Standalone in a Virtual Machine in Azure

- This solution is highly compatible with existing SQL applications that are hosted in an on-premises location. You can migrate the installation of SQL Server from an on-premises location to Azure by uploading the virtual hard disks. Because of this, you can use this solution in scenarios where you want to migrate your existing database workload without modifying it.
- Existing applications with a customized SQL installation benefit from using SQL Server Standalone on a virtual machine in Azure. For example, SQL Server Analysis Services (SSAS) is not available for the SQL Database service. If you host SQL Standalone on a virtual machine in Azure, you can enable the services that are used with SSAS.
- If you want to enable a resilient scenario for your data, you have to manually configure SQL Server AlwaysOn.

One of the most common reasons for hosting SQL Server standalone in Azure is for lift and shift scenarios. Lift and shift workloads are applications that are migrated from one platform to another with minimum changes to the application's source code or configuration. Migrating an existing SQL workload to Azure IaaS is much easier to accomplish than migrating your application to SQL Database. After you migrate, you can analyze your existing SQL workload and determine the degree of compatibility with SQL Database. For new (greenfield) applications, SQL Database provides a near-zero maintenance experience that can accelerate the time to market for the newly created applications.

 **Reference Link:**
https://docs.microsoft.com/azure/sql-database/sql-database-paas-vs-sql-server-iaas

Lesson 2
Managing SQL Databases in Azure

The Management Portal provides convenient ways to create and manage databases. There is also a unique portal that you can use to manage the tables, views, and other objects for your SQL database.

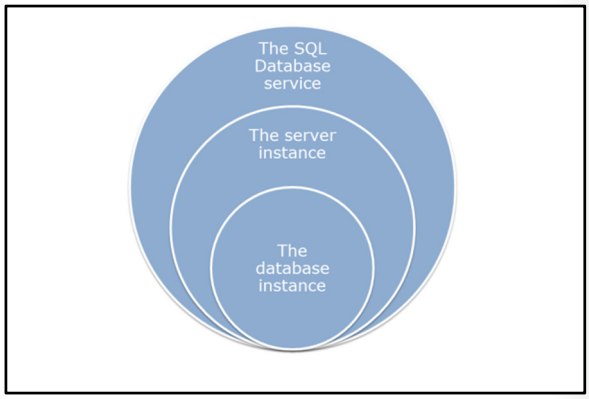
This lesson lists the various online management tools that are available for interacting with SQL databases in Azure.

Lesson Objectives

After completing this lesson, you will be able to:

- Create a server and a database by using the Management Portal.

Creating an Azure SQL Database Instance



You can create a new database instance in the SQL Database service by using the portals, Service Management REST API, Windows PowerShell, or Visual Studio 2013. Later in this lesson, screenshots are provided to show you how to create a database in the Management Portal and a demonstration is included to show you how to create a database in the Azure Preview Portal.

Before creating a database instance, you must create a logical server. This server must have a globally unique name which will be used to connect to the databases by using the TDS protocol. The server will have an administrator account created by using SQL Server authentication. Windows authentication is not currently supported. After you create the logical server, you can create database instances within the service. Database names do not need to be globally unique.

Creating New Databases by Using the Management Portal

To logically group your databases, you can create a server by using the Management Portal:

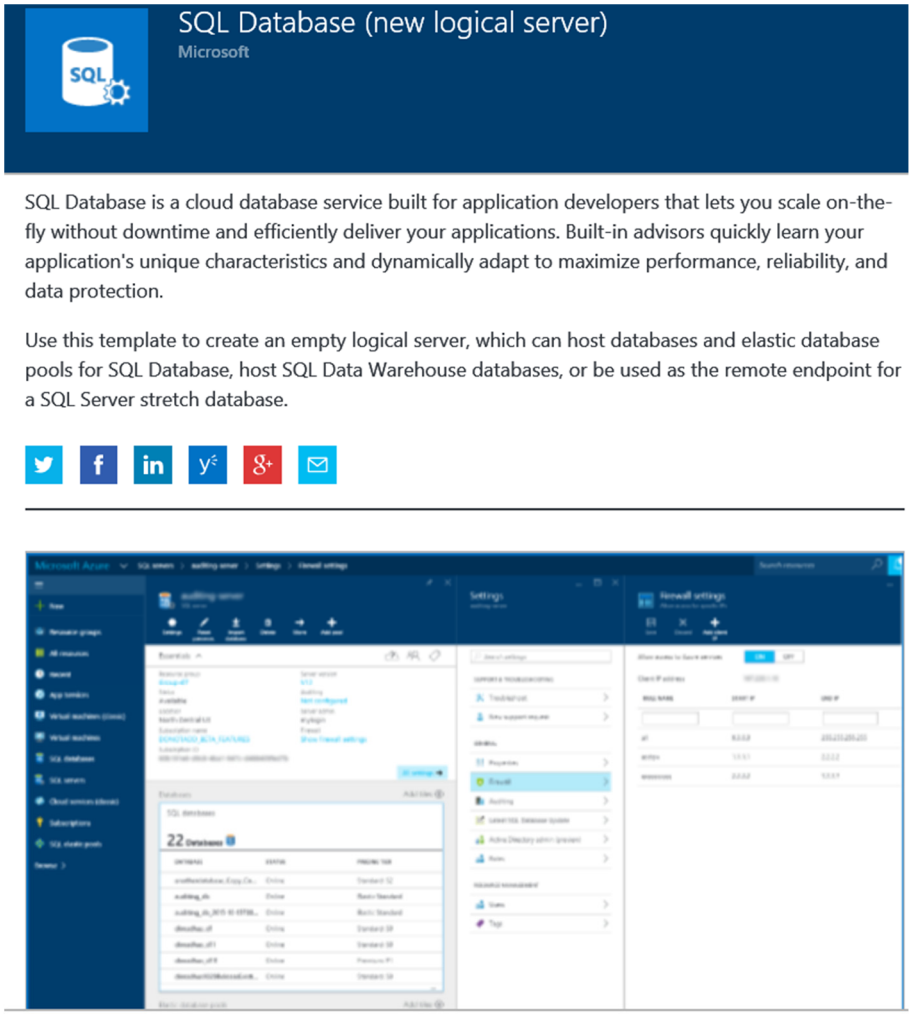


FIGURE 4.2: NEW SQL SERVER

You also can create databases by using through the Management Portal. When you create a database, you can place it in an existing server or create a new server:

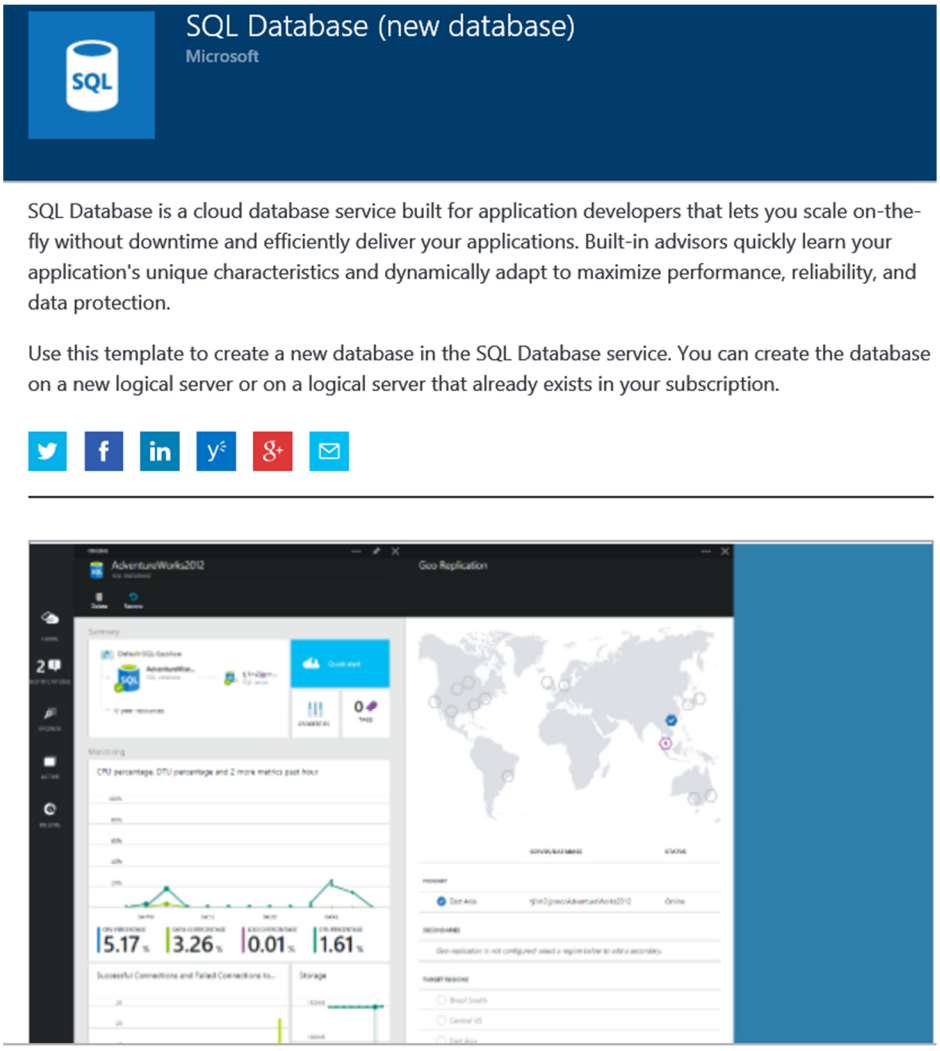


FIGURE 4.3: NEW SQL DATABASE

Management Portal – SQL Database

A unique Silverlight-based portal for your SQL Database instances is available in the current Management Portal. You can use this portal to view some simple analytical data for your databases and to manage database objects either visually or through T-SQL queries.

Azure SQL Database Management Portal:

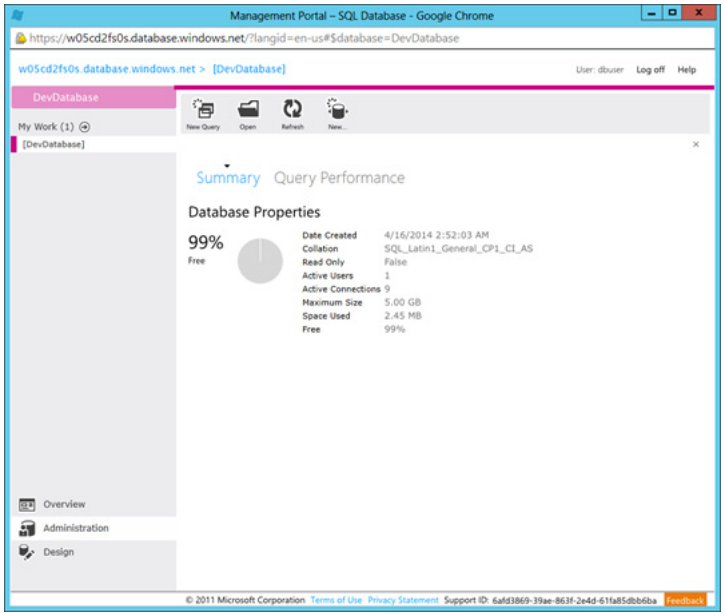


FIGURE 4.4: AZURE SQL DATABASE MANAGEMENT PORTAL

Lesson 3
Azure SQL Database Tools

One of the advantages of SQL databases in Azure is the ability to use many monitoring tools that you use for on-premises databases.

This lesson describes the existing management tools and how you can use them to manage SQL databases in Azure.

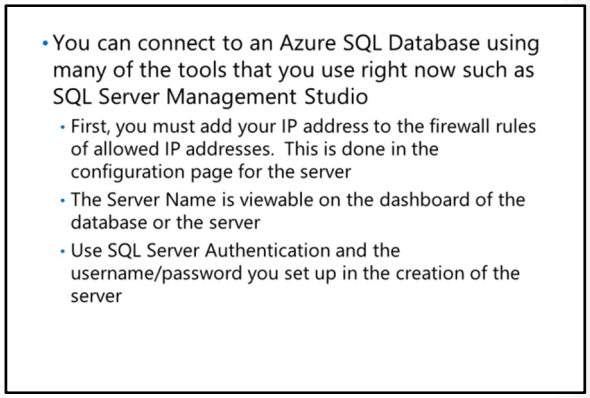
Lesson Objectives

After completing this lesson, you will be able to:

- Manage your SQL Database instances by using SQL Server Management Studio.
- Manage your SQL Databases instances using Visual Studio's tools.

SQL Server Management Studio

One of the greatest benefits of using the TDS protocol is that it provides many tools for connecting to SQL databases and servers. Because the protocol is fully supported by the SQL Database service, the client tools and applications do not need to be concerned about the actual architectural implementation. This enables scenarios where you can use a local SQL installation (such as LocalDb) for development and testing a SQL database for production. You can make this transition by swapping connection strings as appropriate.



Setup

When you configure your SQL Database server instance, you are required to specify a user name and password. This login is the server-level principle for your SQL Database server. The login is similar to the sa principal in SQL Server Standalone. At any time, you can use the master database that is created with your server to manage logins and roles for your databases. You can create additional logins by using standard T-SQL queries.

You can use the CREATE LOGIN, ALTER LOGIN, or DROP LOGIN statements to manage logins in SQL Database.

Managing Logins

```
CREATE LOGIN login1 WITH password='<ProvidePassword>';
```

After you configure your server, you must also configure the following access rules:

- **Firewall**. You can configure the firewall to allow access to your SQL database from a pre-defined list of IP address ranges. Any IP address that is not included in these ranges will not be able to connect to the TDS endpoint.
- **Azure Service Access**. You can configure a Boolean configuration option to indicate whether other Azure services can access the TDS endpoint of your SQL database.

Visual Studio

Visual Studio provides several ways to manage SQL Server standalone and SQL Database instances. You can use the Server Explorer pane to connect to and manage a SQL database. You also can use the Server Explorer pane to manage various Azure services including SQL Database. You can use the SQL Server Object Explorer to manage both individual databases and an entire server. You can use Visual Studio database projects to design a database in a declarative way and publish the resulting script. You can modify the options for these projects so that the script that is generated is compatible with the SQL Database service.

SQL Server Management Studio

A TDS endpoint is exposed for each logical server in SQL Database. This allows you to use SQL Server Management Studio with SQL Database in the same way you will use it with SQL Server standalone.

Migration Tools

SQL Migration Scripts

You can generate migration scripts by using the Generate Scripts task in SQL Server Management Studio.

- You can copy the script file that is generated to the Clipboard, open it in a script window, or save it to a file.
- A simple method that is highly compatible with different versions of SQL Server.
- The scripts can take a while to run and require you to use a tool to run them against the target database.

- You can use SQL Server Integration Services to define a migration plan for on-premises databases
- SQL Database Migration Wizard analyzes your existing database and generates a script (and bulk copy files) to migrate your database
- Azure Websites Migration Assistant uses SQL Management Objects to analyze your existing database and migrate it

Importing and Exporting Data-Tier Applications

You can export the data-tier applications from a SQL database and import them into another.

- A data-tier application contains the definitions of the objects in the database.
- The application package also contains the data that is stored in the different database objects.
- You can import data-tier applications into a SQL database in Azure.

Many new tools are released to analyze your existing databases, create a migration strategy, and then eventually perform the migration operation. These tools include the SQL Azure Migration Wizard and the Azure Websites Migration Wizard.

SQL Server Integration Services (SSIS)

You can use SSIS to both plan and perform migration of data from a SQL Server standalone installation to a SQL database in Azure. This is ideal for scenarios where the existing database uses the functionality that is not supported in the SQL Database service. You can migrate or remap the schema and data into a set of objects that can be hosted in the managed service.

Reference Link: https://docs.microsoft.com/sql/integration-services/sql-server-integration-services

Microsoft Azure SQL Database Migration Wizard

This migration wizard, also known as SQLAzureMW, is designed to analyze existing databases and eventually perform a migration to SQL Database. The tools support a large range of SQL versions starting from SQL Server 2005. The actual migration is performed by generating a script to replicate the schema in SQL Database, and then using the BCP utility to copy the data to the destination from the data files that are exported from the origin.

Reference Link: https://docs.microsoft.com/sql/dma/dma-overview

Reference Link: https://docs.microsoft.com/sql/tools/bcp-utility

Azure Websites Migration Assistant

The Azure Websites Migration Assistant is designed to migrate IIS applications from on-premises to Azure. This tool also migrates SQL databases that are associated with the IIS website by using connection strings. The migration is performed by generating scripts using SQL Server Management Objects.

Reference Link: https://azure.microsoft.com/downloads/migration-assistant/

Lesson 4

Securing and Recovering an Azure SQL Database Instance

Both SQL Server standalone and SQL Database offer a comprehensive set of replication and disaster recovery options.

This lesson describes the high availability features for both SQL Server standalone and SQL as a Service in Azure.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the geo-replication options for SQL databases in Azure.
- Describe the high availability functionality for SQL Server.

Recovery Options for Azure SQL Database

Business continuity describes the planning and execution of maintaining running business operation when an unforeseen event occurs. Azure SQL Database offers many business continuity features with your databases that you would normally have to plan, design and implement. The Azure SQL Database infrastructure also uses replication to implement basic failover for your application within its own infrastructure.

- You can count on infrastructure redundancy in the Azure data centers for the most basic scenarios
- SQL Database also provides it's own set of options for HADR
  - Built-in replicas
    - Transactions are not considered committed to the database until they are written to the target DB, one primary replica, and two secondary replicas
  - Backup and restore
    - Allows you to protect against errant transactions
    - Database is backed up as a whole and can be recovered through the portal
    - The retention period (in days) of your backup is based on your selected service tier

	Basic	Standard	Premium
Point-in-time restore	Any restore point within the past seven days	Any restore point within the past 14 days	Any restore point within the past 35 days
Geo-restore	Maximum downtime < 24 hours	Maximum downtime < 24 hours	Maximum downtime < 24 hours
Standard geo-replication	Not included	Maximum downtime < 2 hours	Maximum downtime < 2 hours

There are three primary recovery options available for SQL Database.

Point-in-Time Restore

Point-in-time restore returns your database and its data to an earlier point in time. The tier of your database determines how far back you can restore your database.

Database Copy

The database copy feature creates a one-time copy of your database to a new instance in another datacenter. When the operation is complete, the duplicate database is transactionally consistent with the source database. Further transactions against the source database are not copied.

The Import and Export Service

This service allows you to create a copy of your database in the .BACPAC format. This file contains a copy of both the data and schema of a database. This file can then be used to import a database into any other Azure SQL Database instance or a SQL Server installed standalone on a machine.

Geo-Restore

Geo-restore behaves in a similar manner to point-in-time restore. The primary difference is that geo-restore replicates the database to a datacenter in another region. Geo-restore is typically used when there is an outage in a specific Azure region.

Standard Geo-Replication

Geo-replication automates the failover of you primary SQL Database instance to another region. The replica created is not available and cannot accept queries or incoming connections using the TDS endpoint. All transactions to your primary database are replicated to the secondary in an asynchronous manner. In the event of an outage, the primary database is flagged as having degraded service. You can then use automation or application logic to connect to your secondary database.

Reference Link: https://docs.microsoft.com/azure/sql-database/sql-database-business-continuity

Azure SQL Databases Geo-Replication

- Active geo-replication is available for Premium SQL Database instances
  - This feature is asynchronous by default and guarantees that replicas will be eventually consistent
  - You can replicate transactions to as many as four copies of the database
  - Replicas can exist in different regions for geo-redundancy
- You can use the replica of the database as a read-only data source in load-balancing scenarios
  - Example: An application uses the primary database for line-of-business functionality and the replica for reports

	Basic	Standard	Premium
Active geo-replication	Not included	Not included	RTO* < 1 hour RPO† < 5 minutes

Active Geo-replication

Out-of-the-box, active geo-replication offers the same functionality as standard geo-replication for a simple failover scenario. Active geo-replication goes even further and can maintain up to four copies of your database that are replicated in a synchronous (continuous) manner. The secondary databases are readable so that your data is still available in an uninterrupted manner while you conduct your failover orchestration.

Reference Link:
https://docs.microsoft.com/azure/sql-database/sql-database-geo-replication-overview

Lesson 5
Additional Managed Database Services

This lesson introduces the Azure Database services for MySQL and PostgreSQL and then compares the database services to the previously introduced Azure SQL Database service.

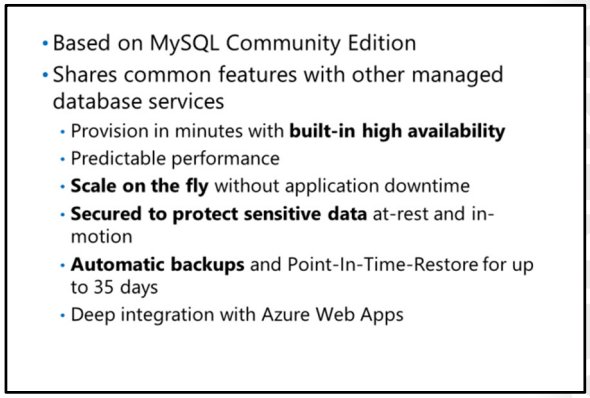
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure Database for MySQL Service.
- Describe the Azure Database for PostgreSQL Service.
- Migrate content from a PostgreSQL database to an Azure Database for PostgreSQL managed instance.
- Integrate Azure Database for MySQL with an Azure Web App.

Azure Database for MySQL

Azure Database for MySQL is a relational database service in the Microsoft cloud based on the MySQL Community Edition database engine. Azure Database for MySQL offers:



- Built-in high availability with no additional cost.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Scale on the fly within seconds.
- Secured to protect sensitive data at-rest and in-motion.
- Automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance.

These capabilities require almost no administration and all are provided at no additional cost. They allow you to focus on rapid app development and accelerating your time to market rather than allocating precious time and resources to managing virtual machines and infrastructure. In addition, you can continue to develop your application with the open source tools and platform of your choice to deliver with the speed and efficiency your business demands, all without having to learn new skills.

Driver and Tool Compatibility

Azure Database for MySQL uses the world's most popular community edition of MySQL database. Therefore, it is compatible with a wide variety of programming languages and drivers. The goal is to support the three most recent versions MySQL drivers, and efforts with authors from the open source community to constantly improve the functionality and usability of MySQL drivers continue.

A list of drivers that have been tested and found to be compatible with Azure Database for MySQL 5.6 and 5.7 is provided below:

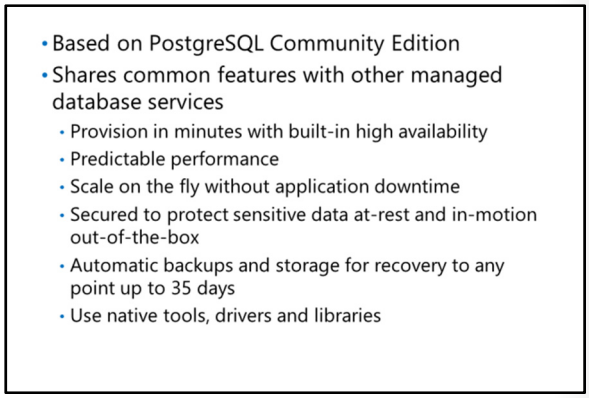
- PHP Driver >= 5.5
- .NET Driver >= 0.27
- Nodejs Driver >= 2.15
- GO Driver >= 1.3
- Python Driver >= 1.2.3
- Java Driver >= 1.6

The compatibility advantage extends to database management tools as well. Your existing tools should continue to work with Azure Database for MySQL, as long as the database manipulation operates within the confines of user permissions. Three common database management tools that have been tested and found to be compatible with Azure Database for MySQL 5.6 and 5.7 are listed below:

- MySQL Workbench >= 6.x
- Navicat >= 12
- PHPMyAdmin >= 4.x

Azure Database for PostgreSQL

Azure Database for PostgreSQL is a relational database service in the Microsoft cloud built for developers based on the community version of open source PostgreSQL database engine. Azure Database for PostgreSQL offers the same features as Azure Database for MySQL:



- Built-in high availability with no additional cost.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Scale on the fly within seconds.
- Secured to protect sensitive data at-rest and in-motion.
- Automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance.

All those capabilities require almost no administration, and all are provided at no additional cost. These capabilities allow you to focus on rapid application development and accelerating your time to market, rather than allocating precious time and resources to managing virtual machines and infrastructure. In addition, you can continue to develop your application with the open source tools and platform of your choice, and deliver with the speed and efficiency your business demands without having to learn new skills.

PostgreSQL Extensions

PostgreSQL provides the ability to extend the functionality of your database using extensions. Extensions allow for bundling multiple related SQL objects together in a single package that can be loaded or removed from your database with a single command. After being loaded in the database, extensions can function as do built-in features.

PostgreSQL extensions must be installed in your database before you can use them. To install a particular extension, run the **CREATE EXTENSION** command from psql tool to load the packaged objects into your database. Azure Database for PostgreSQL currently supports a subset of key extensions as listed below:

- chkpass
- citext
- cube
- hstore
- isn
- ltree
- earthdistance
- fuzzystrmatch
- intarray
- pgcrypto
- pg_partman
- pg_trgm
- tablefunc
- uuid-ossp
- dict_int
- unaccent
- btree_gin
- btree_gist
- plpgsql
- pg_buffercache
- pg_prewarm
- pg_stat_statements
- pgrowlocks
- pgstattuple
- postgres_fdw
- PostGIS
- postgis_topology
- postgis_tiger_geocoder
- postgis_sfcgal

- address_standardizer
- address_standardizer_data_us
- pgrouting

Extensions beyond the ones listed are not supported. You cannot create your own extension with Azure Database for PostgreSQL service.

Lab: Storing Event Data in Azure SQL Databases

Scenario

Now that the web application is ready for publishing, you can begin migrating the web application to Azure by creating a database in Azure. You decided to use a database initializer and Entity Framework Code First to automate the creation of your database in SQL Database.

Objectives

After you complete this lab, you will be able to:

- Create an Azure SQL Database server and a database instance by using the Management Portal.
- Use Entity Framework Code First to initialize and seed a database in the cloud.
- Use the Visual Studio to view live data in the cloud.

Lab Setup

Estimated Time: 45 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Creating an Azure SQL Databases Instance

Exercise 2: Using Entity Framework with Local SQL Server

Exercise 3: Using Entity Framework with Azure SQL Databases

Question: What are some scenarios where it is appropriate to use seed data from an ORM framework in your SQL database in Azure?

Module Review and Takeaways

In this module, you were introduced to Azure Storage and the three basic storage types available. You were also shown examples of how to use the Azure Storage SDK to interact with Table Storage. By using Table Storage, you can design your application today knowing that your storage mechanism is ready to scale to massive levels.

SQL Database provides a service offering for relational databases. It is built to provide the highest level of compatibility with the existing management tools and SSDT in Visual Studio 2013. By using Entity Framework, you can write your code once and rely on the configuration changes to point your ORM to the right database depending on your environment.

Best Practice

You can use configuration settings and web.config transformations to point your application to a different database depending upon the build definition or environment. This allows you to automate your build and test processes with the least amount of code customizations for each environment.

Review Question

Question: When you implement database sharding, why is the federation distribution key important? How does this key help drive performance?

Module 5

Designing Cloud Applications for Resiliency

Contents:

Module Overview

As a developer, you should keep in mind certain considerations while designing applications for the cloud. Although there are many platform improvements available in the ASP.NET ecosystem, you need |to rethink the way you design your applications, and the patterns that are used, with respect to the scalability and reliability metrics present for the cloud applications. Lesson 1, "Application Design Practices for Highly Available Applications," discusses some of the considerations that are needed when you design applications that are hosted in the cloud such that they result in minimal downtime. Lesson 2, "Application Analytics," demonstrates the Application Insights service. Lesson 3, "Building High-Performance Applications by Using ASP.NET," describes the changes in the ASP.NET stack in .NET 4.5 that improve the framework's performance in web applications. Lesson 4, "Common Cloud Application Patterns," introduces a small set of example patterns from the MSDN cloud patterns reference. Lesson 5, "Caching Application Data," compares the Microsoft Azure Cache and Microsoft Azure Redis Cache services.

Objectives

After completing this module, you will be able to:

- Describe the Valet Key, Retry and Transient Fault Handling Patterns.
- Use Load Balancing in a geographically redundant application.
- Create modular applications with partitioned workloads.
- Build High Performance ASP.NET Web Applications.

Lesson 1
Application Design Practices for Highly Available Applications

This lesson describes some of the common considerations that are needed for designing highly available applications.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how to split your application into units of work.
- Describe queue and load balancing strategies for applications.
- Describe the Transient Fault Handling pattern.

Partitioning Workloads

A modular application is divided into functional units, also referred to as modules, which can be integrated into a larger application. Each module handles a portion of the application's overall functionality and represents a set of related concerns. Modular applications make it easier to design both current and future iterations of your application. Existing modules can be extended, revised or replaced to iterate changes to your full application. Modules can also be tested, distributed and otherwise verified in isolation. Modular design benefits are well understood by many developers and architects in the software industry.

- When designing web applications, split your business processes into Partitioning Workloads
- Partitioning Workloads:
  - Can be handled in modular websites, cloud services, or virtual machines
  - Provides the ability to scale the different components of your application in isolation

Load Balancing

Load balancing is a computing concept where the application traffic or load is distributed among various endpoints by using algorithms. By using a load balancer, multiple instances of your website can be created and they can behave in a predictable manner. This provides the flexibility to grow or shrink the number of instances in your application without changing the expected behavior.

- Provide the same service from multiple instances and use a load balancer to distribute requests across all of the instances
- Considerations for selecting a load balancing strategy:
  - Hardware or software load balancers
  - Load balancing algorithms (round robin)
  - Load balancer stickiness
- Load balancing becomes critical even if you have a single service instance as it offers the capability to scale seamlessly

Load-Balancing Strategy

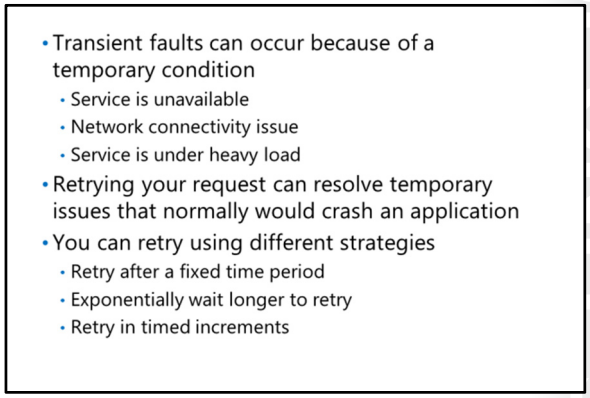
There are a couple of things to consider when choosing a load balancer. First, you must decide whether you wish to use a physical or a virtual load balancer. In Azure infrastructure as a service (IaaS), it is possible to use virtual load balancers, which are hosted in virtual machines, if a company requires a very specific load balancer configuration.

After you select a specific load balancer you need to select a load balancing algorithm. You can use various algorithms such as round robin or random choice. For example, round robin selects the next instance for each request based upon a predetermined order that includes all of the instances.

Other configuration options exist for load balancers such as affinity or stickiness. For example, stickiness allows you determine whether a subsequent request from the same client machine should be routed to the same service instance. This might be required in scenarios where your application servers have a concept of state.

Transient Fault Handling

One of the primary differences between developing applications on-premises and in the cloud is the way you design your application to handle transient errors. Transient errors are as errors that occur due to temporary interruptions in the service or due to excess latency. Many of these temporary issues are self-healing and can be resolved by exercising a retry policy.



Retry policies define when and how often a connection attempt should be retried when a temporary failure occurs. Simply retrying in an infinite loop can be just as dangerous as infinite recursion. A break in the circuit must eventually be defined so that the retries are aborted if the error is determined to be of a serious nature and not just a temporary issue.

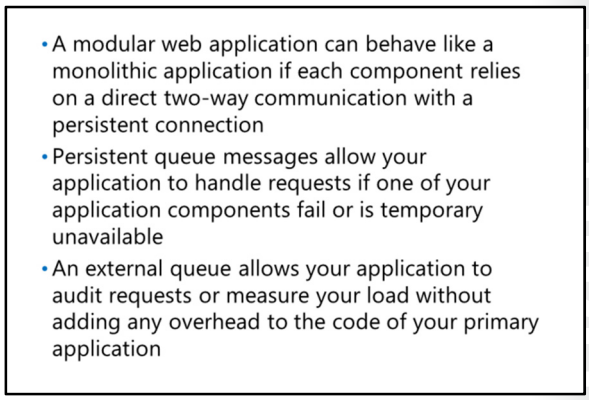
Transient Fault Handling is a pattern that makes your application more resilient by handling temporary issues in a robust manner. This is done by managing connections and implementing a retry policy. This pattern is already implemented in many common .NET libraries such as Entity Framework and the Azure software development kit (SDK). This pattern is also implemented in the Enterprise Library in such a generic manner that it can be brought into a wide variety of application scenarios.

Reference Link: https://docs.microsoft.com/aspnet/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/transient-fault-handling

Reference Link: https://docs.microsoft.com/azure/architecture/best-practices/transient-faults

Queues

Queueing is both a mathematical theory and also a messaging concept in computer science. In cloud applications, queues are critical for managing requests between application modules in a manner such that it provides a degree of consistency regardless of the behavior of the modules.



Applications might already have a direct connection to other application modules using direct method invocation, a two-way service, or any other streaming mechanism. If one of the application modules experiences a transient issue, then this connection is severed and it causes an immediate application failure. You can use a third-party queue to persist the requests beyond a temporary failure. Requests can also be audited independent of the primary application as they are stored in the queue mechanism.

Lesson 2
Application Analytics

Analytics software allow developers and operations the ability to explore the usage and behavior of their web application. With cloud applications, using analytics becomes more important because you have to make scaling and design decisions per iteration.

This lesson describes the Azure Application Analytics service.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Application Insights service.
- Integrate the Application Insights scripts and assemblies with an ASP.NET project to monitor the web application.

Application Insights

Application Insights is not just a monitoring platform but also a set of application extensions that allows you to expose enhanced telemetry for your custom applications. To use Application Insights, you need to add the telemetry application extensions to your existing or new application and then associate the application with an Application Insights instance through configuration. This gives you the flexibility to add telemetry to your existing apps without having to rewrite, or possibly redeploy, your application. Application Insights is also flexible enough to expose telemetry for those applications that are not hosted in Azure.

- Application Insights is a analytics and monitoring service available for your applications
  - View exception stack traces
  - Monitor CPU and resource usage
  - Periodically test URLs from worldwide data centers
  - Monitor usage of your application and most popular requests
- It can be used with .NET or Java
- Applications do not specifically need to be hosted in Azure

At its most basic level, you can use Application Insights to ensure that your application has the intended amount of uptime. You can use a custom dashboard to view various metrics about your application.

This is an example of the Application Insights dashboard:

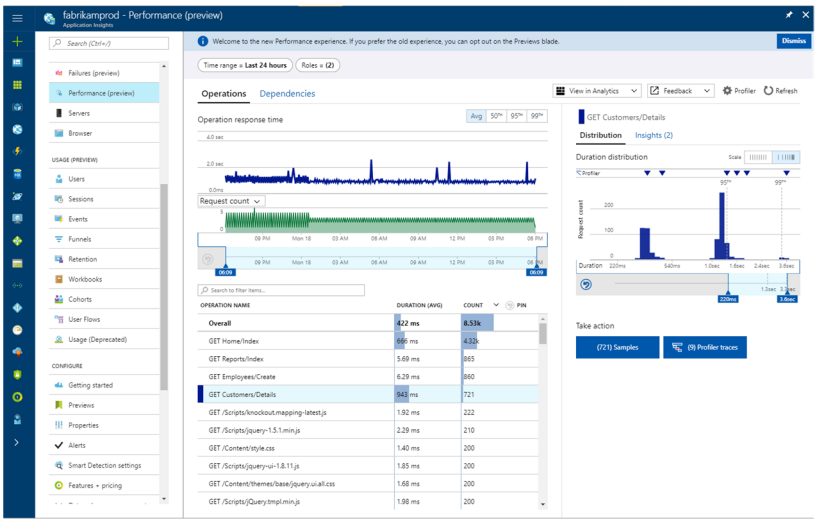


FIGURE 5.1: APPLICATION INSIGHTS DASHBOARD

You can use advanced metrics to determine long-term application behavior and from this behavior analysis you can parse specific performance or application issues. Finally, Application Insights can also give you an insight into the *happy path* for end users. This insight can be gathered by viewing usage telemetry by page or client browser. You can use the insight to determine which areas of your application are conducive for future development investment.

Demonstration: Monitoring a Web Application

 **Note:** To view the latest demo steps, visit the GitHub repository for the course.

Before starting this demo, you must complete the lab in Module 2. For this demo in this module, you will use the available host machine. Also, you must complete the following steps:

1. On the host computer, click **Start**, type **Remote**, and then click **Remote Desktop Connection**.
2. In Remote Desktop Connection, provide the name of your virtual machine in the **Computer** box by using the following format:

   vm20532[Your Name Here].cloudapp.net:[Your VM RDP Port]

 **Note:** The name and port for your virtual machine might be saved in the Computer drop-down list. If this is the case, use this value instead of typing it in manually. If you are unsure about your virtual machine's RDP port, use either of the Azure portals to find your virtual machine's endpoints. The endpoint with the name **Remote Desktop** is the correct port for RDP. This port is randomized to protect your virtual machine from unauthorized access.

3. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.
4. If necessary, sign in by using the following credentials:
   - User name: **Student**
   - Password: **AzurePa$$w0rd**

Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Lesson 3
Building High-Performance Applications by Using ASP.NET

The ASP.NET 4.5 release contained many improvements to the platform that help when building high-performance applications both on premise and in the cloud.

This lesson describes the improvements to ASP.NET for high-performance web applications.

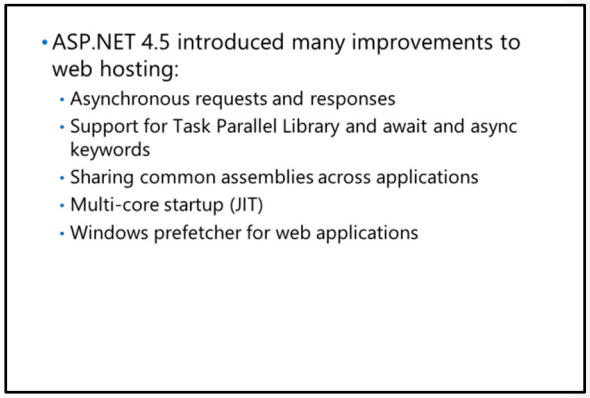
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the asynchronous HTTP modules and handlers.
- Use the **async** and **await** keywords in an ASP.NET application.
- Compare the options for state management in ASP.NET.

Web Hosting Performance Improvements

When compared to their previous versions, ASP.NET 4.5 and .NET 4.5 have a variety of improvements that help you design high-performance applications in the .NET environment.



Performance improvements for Web Hosting
Sharing common assemblies

The Intern functionality enables your operating system to store only a single copy of each assembly for all web applications hosted in IIS. The assemblies in your web application's BIN folder are replaced with a symbolic link to reduce the amount of identical assemblies in memory for the web server.

Using multi-core JIT (just in time) compilation for faster startup

Similar to how the Windows startup moved from single-core to multi-core, JIT compilation for ASP.NET web applications too moved from single-core to multi-core. Modern machines tend to use more processors with less CPU speed per processor which could theoretically slow down JIT compilation. Multi-core JIT compilation is enabled by default and spreads the compilation across all available processor cores.

Prefetching for web applications

Windows prefetcher reduces the startup time for your most commonly used desktop applications by preloading application components before you place a request to start the client application. By using IIS, you can enable prefetcher for ASP.NET web applications through custom configuration options.

Reference Link: https://docs.microsoft.com/aspnet/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/web-development-best-practices

Asynchronous HTTP Modules and Handlers

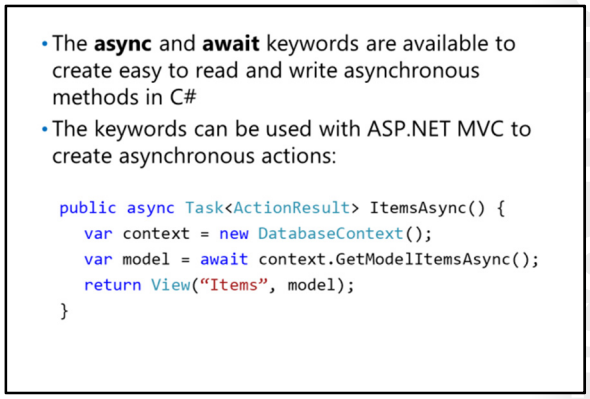
In the past, you could write asynchronous methods by using the Task Parallel Library. This required a deep understanding of the way the HTTP request classes worked with ASP.NET and also some custom configuration. Even then, each request did not release the thread until the completion of all logic, even asynchronous. In ASP.NET 4.5, new methods are added in the ASP.NET stack to read streams and flush asynchronously. Now asynchronous methods will release their threads while they are waiting for the task to be completed. This will ensure that more threads are available across the entire application and thereby increase the performance. This is also useful in scenarios where multiple requests need to be made to services that might introduce latency or transient errors. Normally these requests would lock up multiple threads, but now these threads can be used to service other requests while they are waiting for the external service to respond.

 **Reference Link:** https://docs.microsoft.com/dotnet/standard/parallel-programming/task-parallel-library-tpl

The Async Keyword

.NET 4.5 introduced the **async** and **await** keywords that build upon the task programming concepts introduced in .NET 4.0. The **async** flag is used to identify methods that will use the keywords. The **await** flag is used as shorthand for Task.ContinueWith.

 **Reference Link:** https://docs.microsoft.com/dotnet/standard/parallel-programming/chaining-tasks-by-using-continuation-tasks

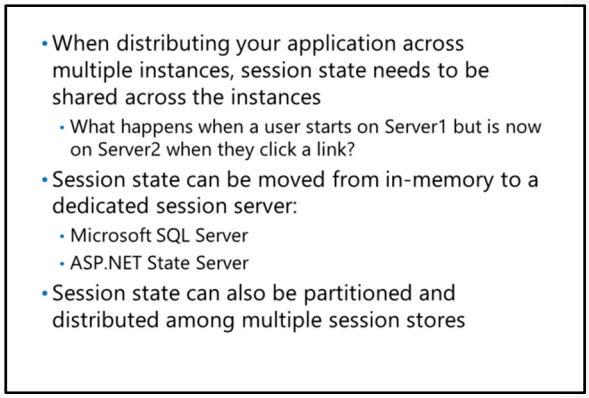


The advantage with the keywords is that they offer more readable code that can easily be parsed by other developers. Traditional asynchronous applications are in general hard to understand, debug, and modify. Typically, you would need large workflow diagrams to understand asynchronous applications developed before the Task Parallel Library and **async** and **await** keywords were introduced. The compiler does the translation from these keywords to their task equivalents.

 **Reference Link:** https://docs.microsoft.com/dotnet/csharp/async

State Management

ASP.NET provides a variety of state options that should be considered when developing your cloud application.



Client-Based State

You can save the application's state on the client-side by using ASP.NET. This is done by using **ViewState**, **ControlState,** or cookies. These options are typically discouraged because of multiple reasons. Firstly, HTML applications are ideally designed to be stateless. Second, these options have a tendency to increase the overhead and payload for responses and requests. Finally, if you want to host multiple instances of your ASP.NET application, then these options require additional configuration because each instance will have different identifying machine data. If the client state is unavoidable, then you should consider options such as hidden input or query strings.

Application or Session State

Application or session state is typically hosted in-process for an ASP.NET application. In the cloud, this becomes a problem because the load balancer might send you to a new application instance that might not have the in-process session values. For example, assume that you have three application instances but have signed in using the first. The load balancer might send you to the second application instance which will not have any state information related to your sign-in. If you require a traditional ASP.NET server state management then consider using a shared state server.

Reference Link: https://docs.microsoft.com/aspnet/core/fundamentals/app-state

Lesson 4
Common Cloud Application Patterns

Although there are many application patterns, some application patterns have emerged with a new generation of cloud-native web applications. MSDN includes a curated list of some of the most common cloud application patterns.

This lesson describes in detail the three examples of the curated cloud application patterns.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Retry pattern.
- Use the Valet Key pattern to access resources.
- Use the Sharding pattern to scale a constrained data storage mechanism.

Cloud Application Patterns

MSDN contains a guide published by the patterns & practices team that provides not just guidance but also over 20 examples of the most common design patterns used for cloud applications. These patterns are neither specific to ASP.NET or Microsoft. In this lesson, you will review a subset of the patterns and discuss example scenarios where they are used. For each pattern, code samples are available in the link below.

- MSDN provides a collection of cloud design patterns
- Patterns:
  - Retry
  - Valet Key
  - Sharding

Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications

Reference Link: https://docs.microsoft.com/azure/architecture/patterns/

The Retry Pattern

Problem: Intermittent Errors with Cloud Services

An application that communicates with elements running in the cloud must be sensitive to the transient faults that can occur in this environment. Such faults include the momentary loss of network connectivity to components and services, the temporary unavailability of a service, or timeouts that arise when a service is busy.

- The Retry pattern is designed to handle temporary failures
- Failures are assumed to be transient until they exceed the retry policy
- The Transient Fault Handling Block is an example of a library that is designed to implement the Retry pattern (and more)
- Entity Framework provides a built-in retry policy implementation
  - Implemented in version 6.0

These faults are typically self-correcting, and if the action that triggered a fault is repeated after a suitable delay it is likely to be successful. For example, a database service that is processing a large number of concurrent requests may implement a throttling strategy that temporarily rejects any further requests until its workload has eased. An application attempting to access the database may fail to connect, but if it tries again after a suitable delay it may succeed.

Solution: Application Logic to Retry Requests That Have Temporarily Failed

In the cloud, transient faults are not uncommon and an application should be designed to handle them elegantly and transparently, minimizing the effects that such faults might have on the business tasks that the application is performing.

If an application detects a failure when it attempts to send a request to a remote service, it can handle the failure by retrying the application logic after a short wait. For the more common transient failures, the period between retries should be chosen so as to spread requests from multiple instances of the application as evenly as possible. This can reduce the chance of a busy service continuing to be overloaded. If many instances of an application are continually bombarding a service with retry requests, it may take the service longer to recover.

If the request still fails, the application can wait again and make another attempt. There should be a limit on attempts to avoid sending endless requests to a service that may actually be completely inoperable. All code that access the remote service should be implemented using a retry policy such as the one described here.

Reference Link: https://docs.microsoft.com/azure/architecture/patterns/retry

The Valet Key Pattern

Problem: Serving Secure Data from Distributed Applications

Client programs and web browsers often need to read and write files or data streams to and from an application's storage. Typically, the application will handle the movement of the data—either by fetching it from storage and streaming it to the client, or by reading the uploaded stream from the client and storing it in the data store. However, this approach absorbs valuable resources such as compute, memory, and bandwidth.

- If your application access a resource on behalf of your clients, your servers take on additional load
- You do not want to make your resources publically available so you are typically forced to have your application validate a client
- The Valet Key pattern dictates that your application simply validates the client and then returns a token to the client. The client can then retrieves the resource directly using its own hardware and bandwidth

Data stores have the capability to handle upload and download of data directly, without requiring the application to perform any processing to move this data, but this typically requires the client to have access to the security credentials for the store. While this can be a useful technique to minimize data transfer costs and the requirement to scale out the application, and to maximize performance, it means that the application is no longer able to manage the security of the data. Once the client has a connection to the data store for direct access, the application cannot act as the gatekeeper. It is no longer in control of the process and cannot prevent subsequent uploads or downloads from the data store.

This is not a realistic approach in modern distributed systems that may need to serve untrusted clients. Instead, applications must be able to securely control access to data in a granular way, but still reduce the load on the server by setting up this connection and then allowing the client to communicate directly with the data store to perform the required read or write operations.

Valet Key Pattern Visually:

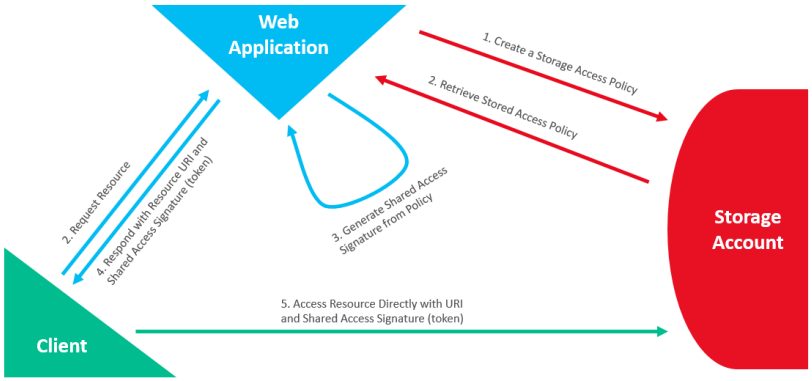


FIGURE 5.2: VALET KEY PATTERN STEP-BY-STEP

Solution: Data Services and Temporary Access Tokens

To resolve the problem of controlling access to a data store where the store itself cannot manage authentication and authorization of clients, one typical solution is to restrict access to the data store's public connection and provide the client with a key or token that the data store itself can validate.

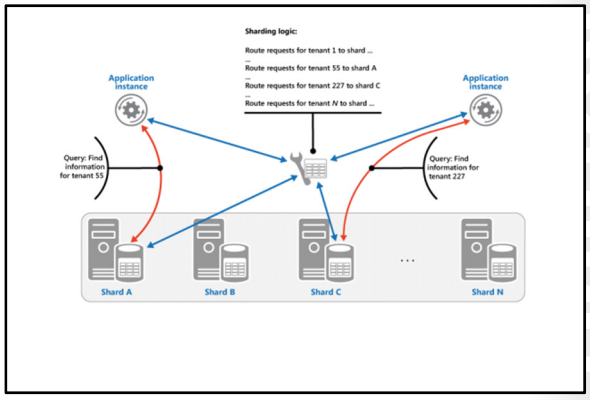
This key or token is usually referred to as a valet key. It provides time-limited access to specific resources and allows only predefined operations such as reading and writing to storage or queues, or uploading and downloading in a web browser. Applications can create and issue valet keys to client devices and web browsers quickly and easily, allowing clients to perform the required operations without requiring the application to directly handle the data transfer. This removes the processing overhead, and the consequent impact on performance and scalability, from the application and the server.

The client uses this token to access a specific resource in the data store for only a specific period, and with specific restrictions on access permissions.

Reference Link: https://docs.microsoft.com/azure/architecture/patterns/valet-key

Sharding Pattern

Problem: Hosting Large Volumes of Data in a Traditional Single-Instance Store



A data store hosted by a single server may be subject to the following limitations:

- **Storage space**. A data store for a large-scale cloud application may be expected to contain a huge volume of data that could increase significantly over time. A server typically provides only a finite amount of disk storage, but it may be possible to replace existing disks with larger ones, or add further disks to a machine as data volumes grow. However, the system will eventually reach a hard limit whereby it is not possible to easily increase the storage capacity on a given server.

- **Computing resources**. A cloud application may be required to support a large number of concurrent users, each of which run queries that retrieve information from the data store. A single server hosting the data store may not be able to provide the necessary computing power to support this load, resulting in extended response times for users and frequent failures as applications attempting to store and retrieve data time out. It may be possible to add memory or upgrade processors, but the system will reach a limit when it is not possible to increase the compute resources any further.
- **Network bandwidth**. Ultimately, the performance of a data store running on a single server is governed by the rate at which the server can receive requests and send replies. It is possible that the volume of network traffic might exceed the capacity of the network used to connect to the server, resulting in failed requests.
- **Geography**. It may be necessary to store data generated by specific users in the same region as those users for legal, compliance, or performance reasons, or to reduce latency of data access. If the users are dispersed across different countries or regions, it may not be possible to store the entire data for the application in a single data store.

Scaling vertically by adding more disk capacity, processing power, memory, and network connections may postpone the effects of some of these limitations, but it is likely to be only a temporary solution. A commercial cloud application capable of supporting large numbers of users and high volumes of data must be able to scale almost indefinitely, so vertical scaling is not necessarily the best solution.

Solution: Partitioning Data Horizontally across Many Nodes

Divide the data store into horizontal partitions or shards. Each shard has the same schema, but holds its own distinct subset of the data. A shard is a data store in its own right (it can contain the data for many entities of different types), running on a server acting as a storage node.

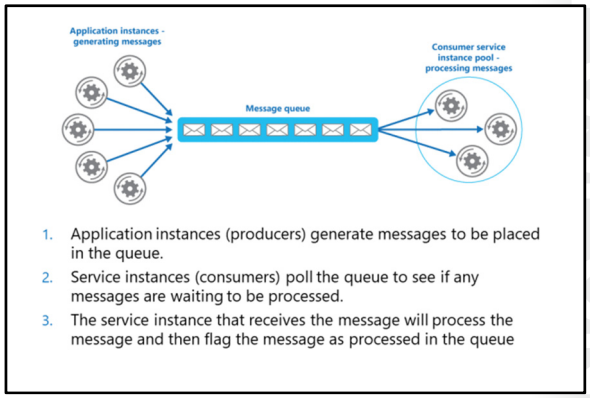
Sharding physically organizes the data. When an application stores and retrieves data, the sharding logic directs the application to the appropriate shard. This sharding logic may be implemented as part of the data access code in the application, or it could be implemented by the data storage system if it transparently supports sharding.

Abstracting the physical location of the data in the sharding logic provides a high level of control over which shards contain which data, and enables data to migrate between shards without reworking the business logic of an application should the data in the shards need to be redistributed later (for example, if the shards become unbalanced). The tradeoff is the additional data access overhead required in determining the location of each data item as it is retrieved.

To ensure optimal performance and scalability, it is important to split the data in a way that is appropriate for the types of queries the application performs. In many cases, it is unlikely that the sharding scheme will exactly match the requirements of every query. For example, in a multi-tenant system an application may need to retrieve tenant data by using the tenant ID, but it may also need to look up this data based on some other attribute such as the tenant's name or location. To handle these situations, implement a sharding strategy with a shard key that supports the most commonly performed queries.

Competing Consumers Pattern

Problem: Handling Variable Quantities of Requests



An application running in the cloud may be expected to handle a large number of requests. The number of requests could vary significantly over time for many reasons. A sudden burst in user activity or aggregated requests coming from multiple tenants may cause unpredictable workload. At peak hours a system might need to process many hundreds of requests per second, while at other times the number could be very small. Additionally, the nature of the work performed to handle these requests might be highly variable.

Using a single instance of the consumer service might cause that instance to become flooded with requests or the messaging system may be overloaded by an influx of messages coming from the application.

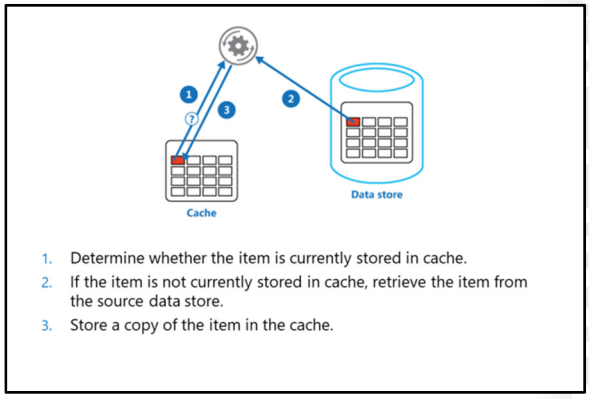
Solution: Asynchronous Messaging with Variable Quantities of Message Producers and Consumers

Rather than process each request synchronously, a common technique is for the application to pass them through a messaging system to another service (a consumer service) that handles them asynchronously. This strategy helps to ensure that the business logic in the application is not blocked while the requests are being processed.

A message queue can be used to implement the communication channel between the application and the instances of the consumer service. To handle fluctuating workloads, the system can run multiple instances of the consumer service. The application posts requests in the form of messages to the queue, and the consumer service instances receive messages from the queue and process them. This approach enables the same pool of consumer service instances to handle messages from any instance of the application.

Cache-Aside Pattern

Problem: Cached Data Consistency



Applications use a cache to optimize repeated access to information held in a data store. However, it is usually impractical to expect that cached data will always be completely consistent with the data in the data store. Applications developers should consider a strategy that helps to ensure that the data in the cache is up to date as far as possible, but can also detect and handle situations that arise when the data in the cache has become stale.

Solution: Read/Write-Through Caching

Many commercial caching systems provide read-through and write-through/write-behind operations. In these systems, an application retrieves data by referencing the cache. If the data is not in the cache, it is transparently retrieved from the data store and added to the cache. Any modifications to data held in the cache are automatically written back to the data store as well.

For caches that do not provide this functionality, it is the responsibility of the applications that use the cache to maintain the data in the cache. An application can emulate the functionality of read-through caching by implementing the cache-aside strategy. This strategy effectively loads data into the cache on demand if it's not already available in the cache.

Lesson 5
Caching Application Data

Azure provides two primary cache mechanisms that help you store consistent data that can be shared by your application's services. Although Redis Cache is now the preferred cache mechanism, it is important to understand Azure Cache because it pertains to existing cloud applications.

This lesson describes the two cache offerings in Azure, Azure Cache and Redis Cache.

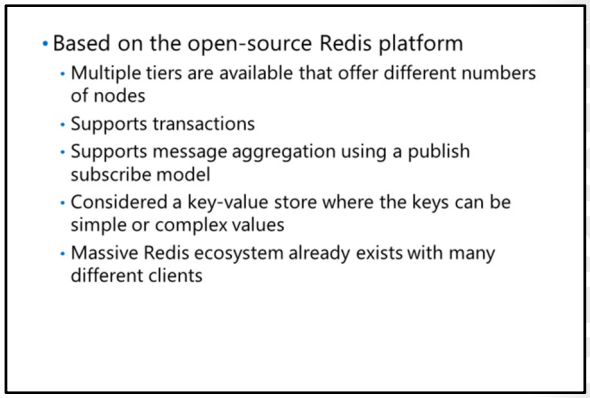
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure Cache.
- Describe the Redis Cache.

Redis Cache

There are two primary cache mechanisms available in Azure, Azure Cache and Redis Cache. Azure cache is deprecated and only exists to support existing cloud applications. All new applications should use the Redis Cache.



Azure Managed Cache

Azure Cache is a managed cache service that is based on the App Fabric platform. You can create the Cache instances by using third-party applications or Windows PowerShell cmdlets. This cache mechanism is typically seen when working with custom cloud service roles.

Redis Cache

Redis Cache is an open-source NoSQL storage mechanism that is implemented in the key-value pair pattern common among other NoSQL stores. Redis Cache is unique because it allows complex data structures for its keys.

Reference Link: https://redis.io/

Azure Redis Cache is a managed service based on Redis Cache that provides you secure nodes as a service. There are only two tiers for this service currently available:

- **Basic**. Single node.
- **Standard**. Two nodes in the Primary/Replica configuration. Replication support and Service Level Agreement (SLA) is included.

Azure Redis Cache provides a high degree of compatibility with existing tools and applications that already integrate with Redis Cache. You can use the Redis Cache documentation that already exists on the open source community for Azure Redis Cache.

Reference Link: https://docs.microsoft.com/azure/redis-cache/

Module Review and Takeaways

In this module, you reviewed some of the patterns and guidance for creating cloud applications. This may not be the first time you would have seen these patterns or practices. When developing disconnected modular web applications running on external hardware such as. Azure, the effort-to-gain ratio of many patterns changes significantly. Many patterns and tools that may be an overkill on premise are now critical with distributed cloud applications.

Review Questions

Question: Why would you want a periodic check or heartbeat of your Azure Website's availability across geographic regions?

Question: How could you implement the Valet Key pattern by using Azure services?

Module 6

Storing Unstructured Data in Azure

Contents:

Module Overview

Many new application workloads require new databases that offer scale and flexibility far beyond the capabilities of a traditional relational database. In Azure, there is a wide variety of NoSQL database services available for applications to store unstructured data in a flexible, schema-free and scalable fashion. Lesson 1, "Azure Storage Overview," introduces the Azure Storage service and details some of the storage types available to applications using Azure Storage. Lesson 2, "Azure Storage Tables," details the Table key-value store available as a NoSQL database in Azure Storage. Lesson 3, "Azure Redis Cache," introduces the Redis Cache key-value based NoSQL store and details how it can be used as a cache database. Lesson 4, "Azure Search," describes the Azure Search service offering that indexes and provides rich-search capabilities for documents stored in structured and unstructured storage. Lesson 5, "Azure Cosmos DB," explores the Azure Cosmos DB service as a flexible NoSQL database that supports a large variety of APIs and models.

Objectives

After completing this module, you will be able to:

- Describe the Azure Storage service.
- Use Azure Search or Cosmos to store NoSQL data.
- Use Azure Redis Cache to store cache data.

Lesson 1
Azure Storage Overview

Microsoft Azure Storage enables you to store unstructured data, files, and messages. You can use Storage as a shared resource to your cloud applications and the various instances of each service.

This lesson provides a brief overview of Storage.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Storage service.
- Identify the types of storage available.
- Describe geo-replication of Storage data.

Azure Storage

With cloud computing, flexible infrastructure is now available for applications that were designed with infrastructure, resilience and scalability in mind. Cloud Services automatically handle the load balancing and infrastructure creation (spin-up) when you wish to scale your application up or down. DocumentDB partitions your data across multiple nodes automatically to grow to handle the amount of data you wish to store.

- The Azure Storage services allow you to store records, files, and simple requests in a flexible, managed, and scalable solution
- Separating the storage of your data from your application allows more flexibility when planning and scaling different aspects of your cloud application scenario(s)
- Storage services are massively scalable which allow you to store large sets of data without being forced to plan partitioning and sharding of your data

Storage is the foundational service that supports all of these flexible services. Storage was designed to make it possible for developers to build large-scale applications by providing a storage mechanism that automatically scales to handle the application's workload. Storage is massively scalable and it is possible to storage large sets of data for data mining, analytical and media workloads.

Storage also has predictable performance and requirements that assists in planning for future growth. For example, operations per second are throttled per blob, disk or storage account. These thresholds are published and can be used to plan for your application's data requirements. You may need to spread virtual machine applications across multiple disks to surpass per-disk Input/Output Operations per Second (IOPS) thresholds. You may also consider spreading media files across multiple storage accounts to surpass per-account IOPS thresholds.

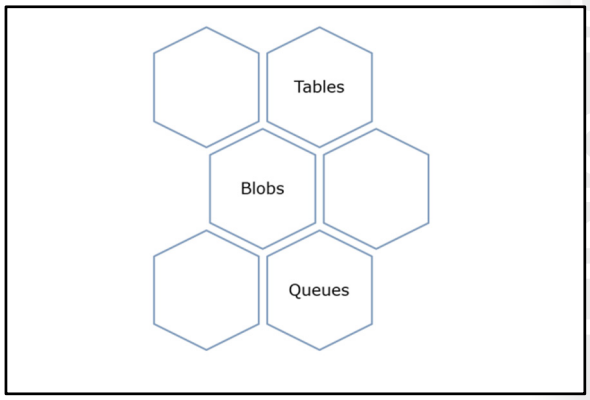
Storage automatically partitions data and load balances your requests among these generated partitions. As your application grows, Storage will automatically allocate the correct quantity of partitions to scale and meet your growing resource requirements.

Storage is also globally accessible using standard HTTP URLs. These URLs can be used with client applications to allow end-users to download multimedia content from your Storage account. Your Storage account can also be protected against unauthorized access by marking resources as private. These private resources can be accessed temporarily if the client application has an accurate security token. This security token is represented as a URL query string parameter to ensure that it is compatible with almost any client device or platform.

Reference Links: https://docs.microsoft.com/azure/storage/common/storage-introduction

Types of Storage

The Storage service consists of four primary types of storage. Each of these are available at unique endpoints as listed below:



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

http://[account].table.core.windows.net/[table]([parition key],[row key])

http://[account].queue.core.windows.net/[queue]

http://[account].file.core.windows.net/[file]

Blob

Storage blobs provide a way to store files so that they can be consumed by other components of your application or client devices. These blobs represent files and can be protected such that tokens are required to access the blob.

Table

The Table service is a NoSQL store that is based on the document paradigm. Each entity consists of a partition key and a row key that together form a unique index. The entities then contain a collection of key-value pairs for the document's attributes.

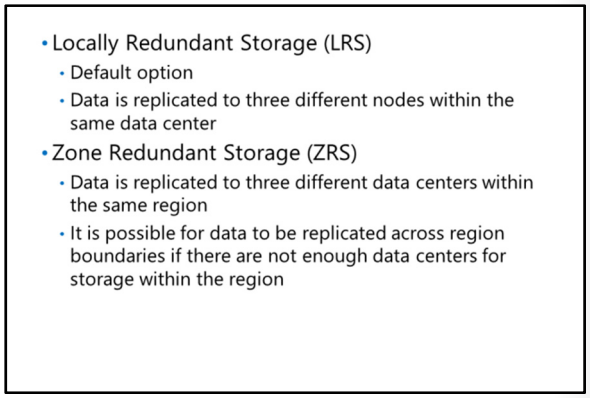
File

File is a service that publishes a Server Message Block (SMB) 2.1 endpoint. This endpoint can be used by virtual machines in Azure in the same way as a shared drive.

Queue

Storage queues are externally managed queues that can persist requests to be consumed by modules in your application. The queue is implemented in a traditional first in, first out (FIFO) pattern. These queues can be measured and requests can also be reviewed (peek) without removing them from the queue.

Geo-Replication in Azure Storage



When an Azure storage account is provisioned, a replication option is selected for the new storage account. The four options affect availability, pricing, and extended functionality. In Azure, geography is broken down into couple of simple concepts. First, the Azure environment consists of multiple regions, which are typically geographically diverse. Second, each region contains one-to-many facilities that are sometimes known as datacenters. These two concepts are important to understand when comparing replication options for Storage. By default, all storage accounts have replicas within their own datacenter and they are organized as triplicates. Different replication levels affect replicas other than the default triplicate.

Locally Redundant Storage (LRS)

This is the simplest and least expensive option. As you know, three copies of your data exist within the same datacenter. If you have data that does not require any replication, then you can use this option and achieve significant cost savings. However, the flip side is that although the data might survive a single physical fault, an outage to the entire datacenter will make this data typically unavailable.

Zone-Redundant Storage (ZRS)

This option replicates your data to another datacenter within the same region. This ensures that your data is available even if there is a fault or outage to an entire datacenter. There are still only three copies, but they are spread out across multiple datacenters in the region.

Note: Some regions only contain one datacenter. In such a scenario, ZRS replicates your data to another datacenter in a different region. For example, in 2014 the Brazil region contained a single datacenter. ZRS replicated data to a Texas, US datacenter. This provided an upgrade to GRS replication without any increase in the cost. However, you need to be careful while using this option because this can have compliance implications.

Geo-Redundant Storage (GRS)

GRS replicates your data to a secondary datacenter in a different region. There will still be three copies in your primary datacenter, but with this option there will be three new copies in the new datacenter. Copies are committed in an asynchronous manner. The primary datacenter will commit the new data and then return a HTTP status message before the secondary datacenter performs a commit. This data pattern is typically referred to as eventually consistent.

Read-Access Geo-Redundant Storage (RA-GRS)

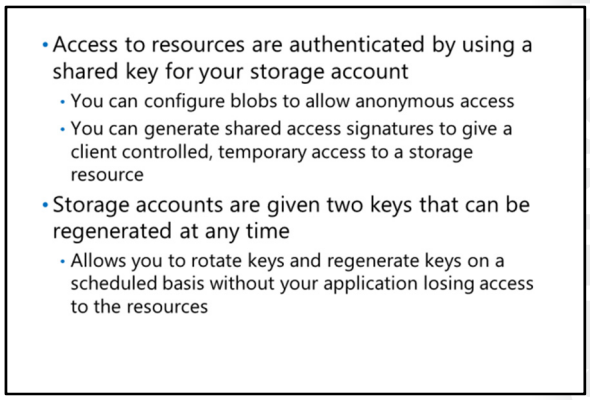
RA-GRS behaves the same way as GRS. The primary difference is that your applications can now access the secondary data in a read-only manner. This is useful for read access during an outage or for distributing your workload across datacenters. For example, your web application can read and write data to the primary datacenter and your reporting application can simply read data from the secondary datacenter.

Reference Link: https://docs.microsoft.com/azure/storage/common/storage-redundancy

Reference Link: https://docs.microsoft.com/azure/storage/blobs/storage-blob-storage-tiers

Accessing Storage Data

The Azure portal does not provide a direct interface for accessing the data that is stored in a storage account. To manipulate storage accounts, you might have to make use of any of the tools provided by the Azure team, such as client libraries, REST API, Windows PowerShell, or Visual Studio's Server Explorer. You can use certain third-party tools, such as LINQPad or the Azure Storage Explorer, to interact with storage accounts.



Reference Link: http://www.linqpad.net/

Reference Link: https://azure.microsoft.com/features/storage-explorer/

Although you can access storage data directly by using the URI, you can individually restrict access or modification, by anonymous users, to the containers and blobs.

Shared Access Signature

A Shared Access Signature is a set of query parameters that can be used by a client to access the resource. The query parameters specify the access level and expiration time, but this data is verified by a signature so it can't be modified manually. Libraries are available to generate the signature in an ad-hoc manner.

Stored Access Policy

Sometimes, signatures need to be generated in a planned manner as opposed to ad-hoc. You can define a Stored Access Policy to create a uniform description that can be used to generate multiple shared access signatures. The policy goes even further and provides a centralized control mechanism for the previously generated signatures. If you revoke the policy, then it will revoke any previously generated signatures. You can use this in scenarios where you need to protect multiple signatures from compromising sensitive data.

Sample application diagram using the Valet Key pattern with Azure Storage:

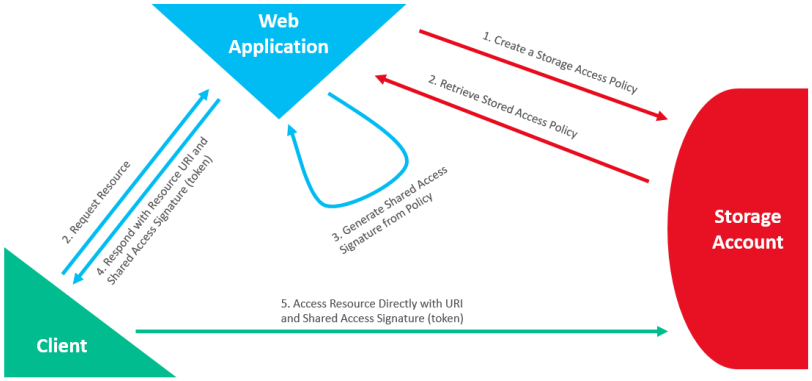


FIGURE 6.1: THE STORAGE VALET KEY PATTERN

Reference Link: https://docs.microsoft.com/azure/storage/common/storage-dotnet-shared-access-signature-part-1

Lesson 2
Azure Storage Tables

The Table service provides a nonrelational database option for storage in Azure. It is tuned to scale to large number of entities and even numerous operations per second. Simply put, the Table service is built with scalability in mind.

This lesson introduces the Table service and describes why it is unique.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Table service.
- Explain the structure of data and partitions in Table storage.
- Describe the relationship between OData and Table storage.

Storage Tables

- Table storage allows you to store flexible datasets that are not constrained by a schema or a model
  - Client applications can dictate the model of the data stored
  - Complex objects of different schema(s) can be stored in the same table
- Built for massive scale (very large datasets)
- Data is partitioned by using a partition key to support load balancing across different nodes
- Data is indexed by a combination of the partition key and a row key for very fast lookup

The Table service provides a NoSQL data storage mechanism to store loosely structured data in a table that you can partition by using a unique partition key. The ideal candidate for Table storage is rarely-modified or immutable data that needs to be stored at a very large scale and accessed very quickly. With traditional relational database management systems, massive data storage is possible but queries are expensive. If you want to access a single item in a table, then you will need a query and this might end up being exponentially more expensive as the size of the table grows. If you use a document-based NoSQL system, such as the Table service, then the problem is solved because it uses a variety of methods including an index and a hash to access a single record in a table. This is how data can be stored at massive volumes while maintaining consistently fast access to specific records.

In the context of the Table service, tables are sets of entities that are related but do not necessarily share the same schema. Each table has a sharding implementation (called *partitions*) where data in the same partition will be on the same server. This means that data in the same partition can be accessed very quickly while the data across partitions typically is accessed much slower. The logical structure for Table storage is as follows:

Storage Account

TableI

ParititonA

Entity1

Entity2

Entity3

PartitionB

Entity4

PartitionC

Entity5

Entity6

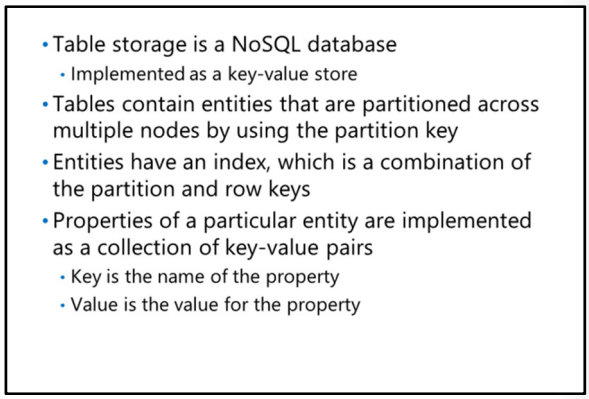
TableII

PartitionD

Entity7

Throughout this lesson, you will explore the structure of Table storage.

NoSQL Data in Storage Tables



The Table service data model can be daunting at first. Tables are composed of partitions. These partitions contain data that is related by the PartitionKey. Each individual entity in a partition has a unique row key. In the table, the partition key and row key must be unique for every entity and they behave in a manner similar to a compound index. Data in the same partition is served from the same physical server, which explains why a query that scans only a single partition is still very efficient. A query that is looking for only a specific entity needs to specify the partition key and the row key. The row key is used like an index and can retrieve an entity without the need for a full table or partition scan.

Partitions can be grouped together in the same server if they are small enough. This is called a partition range. Although this might occur in your application, it is difficult to design around this concept. If you need multiple entities by design, you should always design these entities to share a partition key.

Please refer to Module 5, Designing Cloud Applications for Resiliency for more information about sharding design and principles.

Reference Link: https://docs.microsoft.com/rest/api/storageservices/Understanding-the-Table-Service-Data-Model

Reference Link: https://docs.microsoft.com/rest/api/storageservices/Designing-a-Scalable-Partitioning-Strategy-for-Azure-Table-Storage

Demonstration: Implementing Azure Storage Tables

Note: To view the latest demo steps, visit the GitHub repository for the course.

Before starting this demo, you must complete the lab in Module 2. For this demo in this module, you will use the available host machine. Also, you must complete the following steps:

1. On the host computer, click **Start**, type **Remote**, and then click **Remote Desktop Connection**.
2. In Remote Desktop Connection, provide the name of your virtual machine in the **Computer** box by using the following format:

   vm20532[*Your Name Here*].cloudapp.net:[*Your VM RDP Port*]

Note: The name and port for your virtual machine might be saved in the Computer drop-down list. If this is the case, use this value instead of typing it in manually. If you are unsure about your virtual machine's RDP port, use either of the Azure portals to find your virtual machine's endpoints. The endpoint with the name **Remote Desktop** is the correct port for RDP. This port is randomized to protect your virtual machine from unauthorized access.

3. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.
4. If necessary, sign in by using the following credentials:
   - User name: **Student**
   - Password: **AzurePa$$w0rd**

Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Common Transactions

Although Azure Storage Tables is ideal for data that is not often updated, a full range of CRUD operations are available for Storage table endpoints. You can use the existing client libraries or the REST API endpoints using standard HTTP methods. Querying a table consists of using the **HTTP GET** method with the OData syntax.

- Tables support common transactions for the specified entities:
  - Create
  - Read
  - Update
  - Delete
- Entities behave like **Dictionaries** when updated
  - Any key-value pairs with a new key or added to the collection of properties for an entity
  - Any key-value pairs that re-use an existing key replaces the corresponding value

QUERY	https://[account].table.core.windows.net/[table]()?$filter=[query expression]
GET	https://[account].table.core.windows.net/[table]([partition key], [row key])
PUT	https://[account].table.core.windows.net/[table]([partition key], [row key])
POST	https://[account].table.core.windows.net/[table]
DELETE	https://[account].table.core.windows.net/[table]([partition key], [row key])

MERGE	https://[account].table.core.windows.net/[table]([partition key], [row key])

When updated, entities do not lose key-value pairs that are not specified as part of the object posted.

For example, you have an entity for a student. The students are partitioned by the school they attend and the row key is their student ID. In this particular example, the student has only the first name and last name data.

Partition Key	appleorchardmiddle
Row Key	237548902
First Name	Chris
Last Name	Meyer
Age	11

Assume that you need to update this user in the table named students and in a storage account named cornfielddistrict. You also need to update the Age to 12 and add a new key-value pair for Sport. In such a scenario, the URL and JSON payload you would use is as follows:

PUT: https://cornfielddistrict.table.core.windows.net/students(orchardmiddle,237548902)

JSON Payload

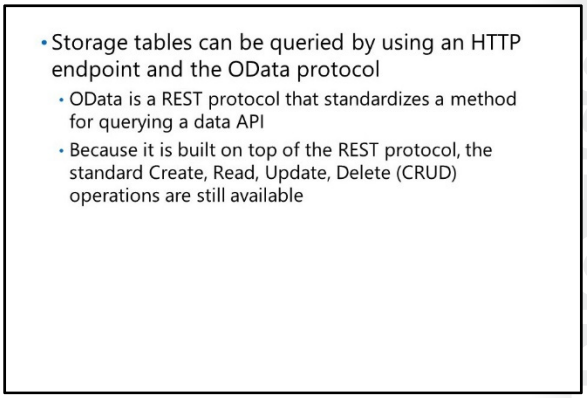
```
{
“Age”: 12,
“Sport”: Tennis
}
```

Your entity will now look like this:

Partition Key	appleorchardmiddle
Row Key	237548902
First Name	Chris
Last Name	Meyer
Age	12
Sport	Tennis

OData Queries

Azure Storage tables can be accessed by using the OData protocol. The OData protocol provides a uniform way to query items in an HTTP endpoint. Traditionally in a RESTful service, the GET endpoint can return a specific item or can return all of the items. The OData query parameters allow you to filter, paginate, or project the result of your request to an HTTP endpoint. The OData protocol can also do a lot more such as typing and batching of requests. There are client libraries available in various programming languages.



Reference Link: http://www.odata.org/

With Storage tables, you can use OData as an easy way to search for particular entities in a table. The **$filter**, **$top** and **$select** query options are currently supported. Results can be returned in either the AtomPub or JSON format. This is specified by including an **Accept** header in your request. Typically, JSON data provides up to a 70% reduction in bandwidth.

If you require a specific entity, the OData protocol provides a mechanism where you can retrieve a single entity in a collection by using the following URL format:

https://[base url]/[resource]([index])

Storage tables deviate from the typical OData protocol by requiring you to specify both the row and partition keys. These form a composite index for an item in Storage tables and are both required to get a specific entity:

https://[account].table.core.windows.net/[table](PartitionKey='[key]',RowKey='[key]')

When using an OData client library, you should be aware that they might not account for this small deviation. The client libraries might also support additional OData query options such as **$skip** and **$expand** that are currently not supported by Storage tables.

OData Endpoints

Standard OData Endpoints are available for all tables stored in Storage Tables. There is a small difference from a traditional OData key lookup. In Table Service, keys are composite and you are required to pass in both the partition key and row key.

- Base URL:
  - https://**[account]**.table.core.windows.net
- Get an entity by a partition and row key:
  - [GET] /**[table]**(**[PartitionKey]**,**[RowKey]**)
- Query a table for entities that match an expression:
  - [GET] /**[table]**()**?$filter=[query expression]**
- Delete an entity:
  - [DELETE] /**[table]**(**[PartitionKey]**,**[RowKey]**)
- Insert or replace an entity:
  - [PUT] /**[table]**(**[PartitionKey]**,**[RowKey]**)

Lesson 3
Azure Redis Cache

Redis Cache is an implementation of the Redis Database engine that is offered as a service on Azure. Using the Redis Cache service, one could create a cache instance quickly using the key-value based store.

This lesson introduces the Redis Cache service as an option for NoSQL caching in Azure.

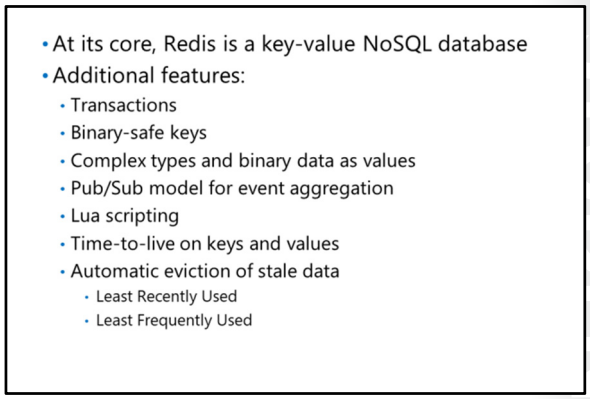
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Redis Cache service.

Redis

Redis is an in-memory key-value based data store that can be used in a wide variety of scenarios. Redis performs at a very high-level using simple strings as the value for its various keys making it optimal as a cache solution. Redis also allows more complex data types to be stored as the value for each key including (but not limited to):



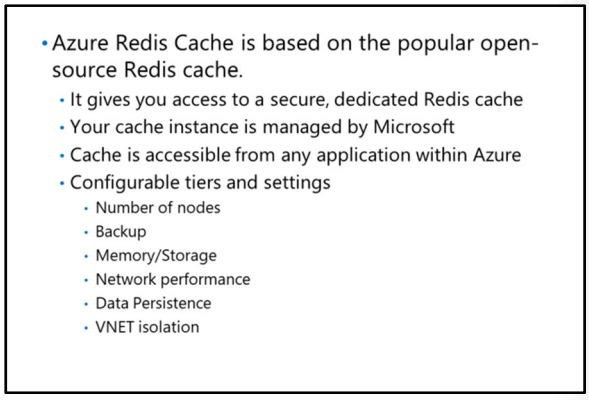
- **lists**: Collection of data.
- **hashes**: Similar to a key-value dictionary of data.
- **geospatial**: Location data

Redis has support for advanced database features including:

- Direct scripting on the database server using the Lua language.
- Transactions that span multiple operations.
- Configurable on-disk persistence.
- Replication across multiple nodes.

Azure Redis Cache

Azure Redis Cache is a cache service that uses the Redis database engine to implement the dedicated cache instances. At the simplest level, you can create a simple Redis Cache instance that is managed by Azure but still accessible using Redis tools and SDKs from various application platforms and environments.



In addition to the typical Redis functionality, the Azure Redis Cache service offers extended features that are available at higher service tiers that differentiate the service from self-hosting the Redis database engine.

These features include:

- An in-portal Redis console experience.
- Snapshots and back-up of data stored in Redis cache.
- Automatic sharding of data across multiple Redis nodes without user configuration.
- VNET deployment options for security and isolation.

Cache Model

At the simplest level, you can store simple string key-value pairs in Redis cache. A key used in Redis Cache is typically a string:

GET setting

EXISTS setting

However, a key in Redis cache is binary-safe and it can be a binary file like a Word document if you like. This may not be ideal as it would be difficult and bandwidth-intensive to query the key. It is recommended that you use simple strings for your keys but you can always use delimiters to indicate how keys are related:

GET setting

GET setting:subsetting

Lua in Redis

For advanced scenarios where you want to create complex scripts, Redis has Lua available as an embedded scripting language. You can evaluate Lua scripts using the following two techniques:

Evaluate a Script File

You can start by writing simple Lua scripts and saving them to .lua files. A simple script that gets the value of the messages:welcome setting would look like this:

return redis.call("GET", "messages:welcome")

This script could then be invoked using the redis-cli tool:

redis-cli --eval script.lua

Ideally, you would not hard-code a string value in the script, so you would pass in the key using a Key parameter. First, you would update your script to look like this:

```
return redis.call("GET", KEYS[1])
```

Then you would pass in the messages:welcome string as a Key argument to the redis-cli tool:

```
redis-cli --eval script.lua messages:welcome
```

EVAL keyword in the console

Alternatively, you can evaluate a script or logic directly in the console using the EVAL command. To evaluate our script.lua file, we would simply invoke the EVAL command in the console like this:

```
EVAL $script.lua 1 messages:welcome
```

You may have noticed that we needed to indicate to the EVAL command how many Key arguments to expect. We would need to do something similar if we wished to skip the script file and invoke our command directly:

```
EVAL "return redis.call('get', KEYS[1])" 1 messages:welcome
```

Lesson 4
Azure Search

Azure Search is a search-as-a-service cloud solution that gives developers APIs and tools for adding a rich search experience over your content in web, mobile, and enterprise applications.

This lesson introduces Azure Search as an option for adding search functionality to an existing application.

Lesson Objectives

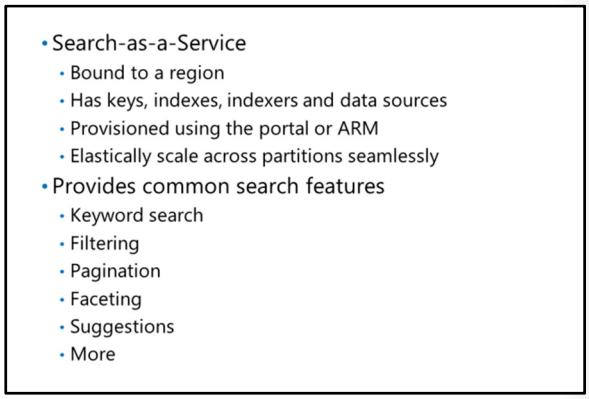
After completing this lesson, you will be able to:

- Describe the features of Azure Search.
- Integrate Azure Search into an existing data solution using an indexer.

Azure Search

Azure Search is a managed search engine offering in Azure that allows you to index data from various data sources and then provide a search engine over the indexed data. Azure Search has many features including:

- Faceted search
- Pagination
- Geospatial search
- Suggestions and Spell-check
- Ranking
- Hit Highlighting



Azure Search has its own built-in query syntax but it can also make use of the Lucene query syntax when searching documents.

One of Azure Search's core differentiators is the ability to create custom linguistic analyzers. Using this feature, you can create analyzers to support your full-text search queries across 50+ languages.

Indexing

Before you can perform your first search, you must first create an index in Azure Search. An index means a couple of unique things within Azure Search:

- An index is the scope used for queries over documents within Azure Search.
- Documents are uploaded, updated and managed within the context of an index.
- Indexes informs the Azure Search engine about the properties (or fields) that are available in the indexes' documents and the capabilities that are appropriate for each document property.

In the simple examples throughout this course we will enable all features for all index fields. In your custom applications, you may choose to tune an index by enabling only specific features over specific fields. Features include:

- **Retrievable**: This means that the field can be retrieved as the result of a search query.
- **Filterable**: This means that the search query can filter using this field. For example, you can filter products which are older than 3 years.
- **Sortable**: This means that the results of a search query can be sorted using this field. For example, you can enable this feature to sort using the price of a product.
- **Facetable**: This means that the field can be used to create grouped metadata about the results. For example, if you enable this for the color field on your products, you will receive results metadata indicating how many results match for each unique value for the color field.
- **Searchable**: This means that a text search can be performed on this field.

Queries

Search Queries can be performed on an Azure Search index using the REST API.

Every REST API request originates from a common base URL:

https://[account].search.windows.net/indexes/[index]/docs

For example, if your account name is **edxdemo** and the index name is **products**, the base URL would be:

https://edxdemo.search.windows.net/indexes/products/docs

Authentication: A *api-key* header is required to use the REST API. This header can be an admin or query key. These are found in the Azure portal.

Query string parameters are used to indicate options for the search. The parameters include (but are not limited to):

- **api-version**: This parameter is the first required parameter and indicates the version of the API you would like to use for your request.
- **search**: This parameter is the second of the only two required parameters. This parameter indicates the actual text-based search query. To return all documents, you can use the ***** (wildcard) operator. You can either use the built-in Simple query syntax or the newer Lucene query syntax with this parameter.
- **searchMode**: This parameter indicates whether or not you want to match on all words in the query or any word in the query. By default, searches match on *any* words.
- **facet**: This parameter indicates that you would like to return faceted metadata about specific fields in your results set. For example, you may wish to see metadata about the price ranges for all prices in every document in your results set.
- **$count**: This OData parameter indicates that you would like to return a number indicating the total number of results. By default, Azure Search will not return all results so this number is useful when figuring out if your user interface needs to show pagination elements.
- **$skip** & **$top**: These OData query parameters are used to return a slice of the result set. For example, you may have 500 results that match your query but you wish to show only 50 results at a time. To do this, you would start on page #1 by skipping *0* results and showing the top *50* results. When the user wants to view page #2, you can skip *50* results and show the next top *50* results. These two parameters are commonly used to implement pagination.
- **$orderby**: This OData parameter sorts the result set using the specified field[s].

- **$select**: This OData parameter allows you to select specific fields that are returned in the result set instead of all available fields. This is useful in scenarios where you want to minimize the amount of unused fields that are returned.
- **$filter**: This OData parameter implements an OData-style filter on the results of the query. For example, you may wish to implement a filter that returns only products that are **Yellow** in color. This can be done using the Color eq 'Black' filter. OData filters are performed together with the search query and can be used to further refine a result set after the search query has been applied.

 **Reference Link:** https://docs.microsoft.com/rest/api/searchservice/simple-query-syntax-in-azure-search

 **Reference Link:** https://docs.microsoft.com/rest/api/searchservice/lucene-query-syntax-in-azure-search

Lesson 5
Azure Cosmos DB

Azure Cosmos DB is an Azure-native database service that focuses on providing a high-performance database regardless of your selected API or data model. Azure Cosmos DB offers multiple APIs and models that can be used interchangeably for various application scenarios.

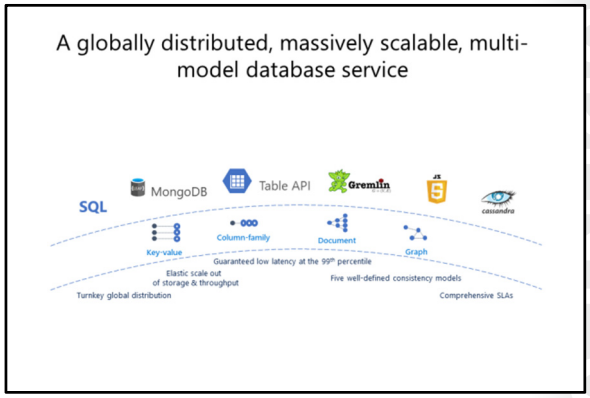
This lesson goes in-depth into the Azure Cosmos DB platform.

Lesson Objectives

After completing this lesson, you will be able to:

- Identify the models and APIs available in Azure Cosmos DB.
- Use the Change Feed to subscribe to database events.
- Use the SQL API to query items in an Azure Cosmos DB container.

Azure Cosmos DB



Azure Cosmos DB was built from the ground up with global distribution and horizontal scale at its core. It offers turnkey global distribution across any number of Azure regions by transparently scaling and replicating your data wherever your users are. Elastically scale throughput and storage worldwide, and pay only for what you need. Azure Cosmos DB provides native support for NoSQL choices, offers multiple well-defined consistency models, guarantees single-digit-millisecond latencies at the 99th percentile, and guarantees high availability with multi-homing capabilities and low latencies anywhere in the world— all backed by industry-leading, comprehensive service level agreements (SLAs).

APIs

Today, Azure Cosmos DB can be accessed using four different APIs. These APIs will be covered as separate modules in this course and are listed below for your convenience:

SQL (DocumentDB) API

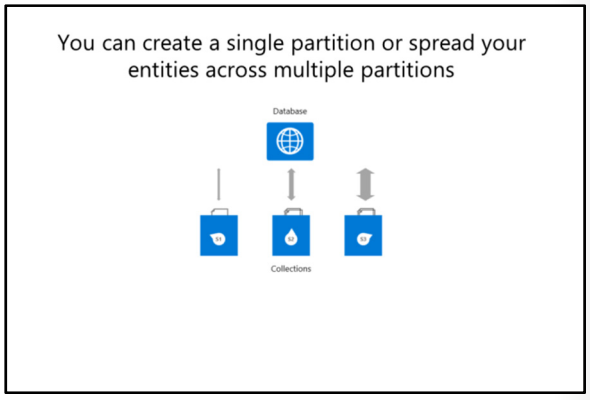
MongoDB API

Graph (Gremlin) API

Tables (Key/Value) API

Partitioning

In Cosmos DB, databases are essentially containers for collections. Collections are where you place individual documents. Each collection is assigned a performance level and that performance level dictates throughput for that collection and its corresponding documents.



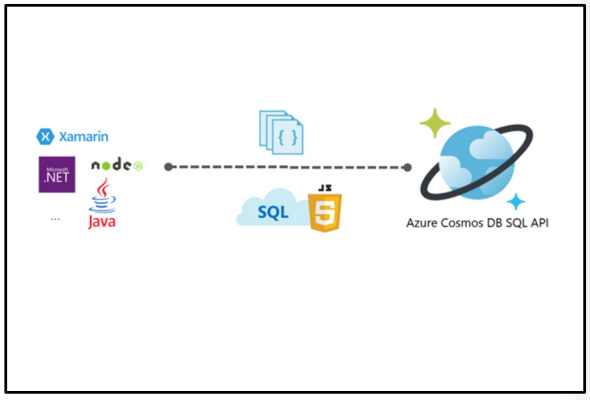
If you have a set of documents that need throughput beyond the limits of an individual collection, you can distribute the documents among multiple collections. Each collection has its own distinct performance level appropriate for its throughput.

If a particular collection is seeing spikes in throughput, you can manage its performance level in isolation by increasing or decreasing the performance level. This change to the performance level of a particular collection will not cause side effects for the other collections. This allows you to adjust to meet the performance needs of any workload in isolation.

You can also scale workloads across collections, if you have a workload that needs to be partitioned, you can scale that workload by distributing its associated documents across multiple collections. The SQL API for Cosmos DB includes a client-side Partition Resolver that allows you to manage transactions and point them in-code to the correct partition based on a partition key field.

SQL API

The SQL API in Azure Cosmos DB is a JavaScript and JSON native API based on the DocumentDB database engine. The SQL API also provides query capabilities rooted in the familiar SQL query language. Using SQL, you can query for documents based on their identifiers or make deeper queries based on properties of the document, complex objects or even the existence of specific properties. The SQL API supports the execution of JavaScript logic within the database in the form of stored procedures, triggers and user-defined functions (UDFs). JavaScript logic can be executed in a transactional manner directly within the database engine.



Resource Hierarchy

The JSON documents stored in Azure Cosmos DB are managed through a well-defined hierarchy of database resources. These resources are automatically replicated to ensure that they are highly available while still allowing you to address each resource using a unique URI.

Lab: Storing Event Registration Data in Azure Storage Tables

Scenario

Even though event registrations could be stored in SQL, you have a unique need. Each event requires a different registration form that can be changed at any time. Essentially, registrations could be of any schema. A relational database such as SQL requires a well-defined schema. Because of your business requirement, you require a database that can store items with flexible structures (or schemas). To facilitate this you have elected to use Azure Cosmos DB for your event registrations.

Objectives

After you complete this lab, you will be able to:

- Use the Azure DocumentDB SDK to create a Cosmos client using the SQL API.
- Use the Azure DocumentDB SDK to create a collection.
- Use the Azure DocumentDB SDK to add documents to a collection.
- Use the Azure DocumentDB SDK to query documents.
- Use the Azure Portal to view Cosmos documents.

Lab Setup

Estimated Time: 45 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Populating the Sign-In Form with Registrant Names

Exercise 2: Updating the Events Website to use Azure Cosmos DB

Exercise 3: Verifying that the Events Website Is Using Azure Cosmos DB for Registrations

Module Review and Takeaways

In this module, you were introduced to Storage and the three basic storage types available. You were also shown examples of how to use the Azure Storage SDK to interact with Table storage. You also learned how Azure Cosmos DB and Azure Redis Cache can be utilized within your cloud-based applications.

Best Practice

You can effectively use configuration settings and Web.config transformations to point your application to a different database depending upon the build definition or environment. This enables you to automate your build and test processes with the least amount of code customizations for each environment.

Review Question

Question: When implementing database sharding, why is the federation distribution key important? How does this key help drive performance?

Module 7

Storing and Consuming Files from Azure Storage

Contents:

Module Overview

When you want to scale to different cloud instances, storing files to a local disk becomes a difficult process to maintain and eventually an unreliable method of storage. Azure provides a Blob storage mechanism that not only offers high performance but also supports integration to Microsoft Azure Content Delivery Network (CDN) for low latency downloads. Lesson 1, "Azure Storage Blobs," describes the Blob service and the types of blobs supported. Lesson 2, "Controlling Access to Storage Blobs and Containers," provides details on the ways that you can secure and grant temporary access to blobs or containers. Lesson 3, "Configuring Azure Storage Accounts," looks at some of the unique configuration options available for Storage blobs. Lesson 4, "Azure Files," briefly introduces the Azure Files service.

Objectives

After completing this module, you will be able to:

- Describe the Blob service in Microsoft Azure Storage.
- Identify the software development kit (SDK) libraries, namespaces, and classes that are available for blobs.

Lesson 1
Azure Storage Blobs

Blob storage provides a file and data storage mechanism in Azure. It is designed for speed and convenience and also supports a CDN option.

This lesson introduces the Blob service and its basic concepts.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Blob service.
- Explain the structure of the containers and blobs in the Blob service.

Storage Blobs

Storage blobs are a unique component of the Storage service that allows you to store various files needed by your applications in a reliant and scalable manner. All stored blobs are associated with an HTTP URL which allows these blobs to be accessed from any location, platform or device. The blob service was designed to store large amounts of unstructured data or files. These files can be exposed publicly using URLs or protected from anonymous access using SAS tokens.

- You can use Blob storage to store large amounts of unstructured text or binary data.
- You can use Blob storage to store files such as:
  - Virtual hard disk drives
  - Videos
  - Images
  - Log text files
- You can use Blob storage to group blobs into logical, hierarchical containers
- You can secure blobs and make them available for anonymous access

Common Uses of Blob Storage Include:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Performing secure backup and disaster recovery.
- Storing data for analysis by an on-premises or Azure-hosted service.

The Blob service contains the following components:

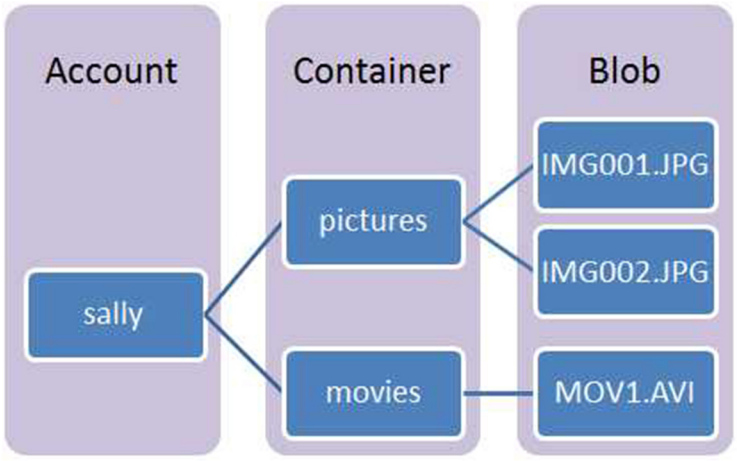


FIGURE 7.1: STORAGE BLOBS HIERARCHY

- **Storage account**. Storage accounts are the highest-level units of encapsulation for objects in storage. A storage account can contain blobs, queues, tables or file shares.
- **Container**. A container represents a group of blobs with shared security scope. Blobs must exist in a single container. Containers cannot be nested.
- **Blob**. Blobs represent individual files in a container. There are two primary types of blobs, *Page* and *Block* blobs.
- **URL format**. Blobs are accessible using the following URL format:

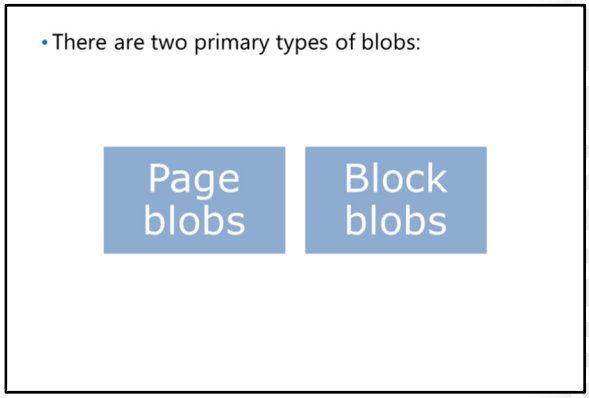
  http[s]://<storage account>.blob.core.windows.net/<container>/<blob>

You can use the following example URL to address one of the blobs in the diagram above:

http://sally.blob.core.windows.net/movies/MOV1.AVI

Blob Types

The Storage containers can contain two types of blobs: block and page blobs. When creating the blob, the blob type must be specified. After the blob is created, the blob type is permanent, and you can only update the properties and content of the blob. For example, you can write a block or set of blocks to a block blob or write pages to a page blob.



Block Blobs

Block blobs are optimized for uploading large files in an efficient manner. Block blobs are composed of sets of blocks that are uniquely identified using a block ID. Block blobs are modified by writing a set of blocks. The set of blocks are then committed using a commit operation and the block IDs. After you upload one or more blocks to your block blob, they are associated with the blob but not committed. You must then commit the set of blocks

to the block blob by again using their block IDs. Uncommitted blocks can also optionally be discarded. Block blobs are typically used for most files and multimedia.

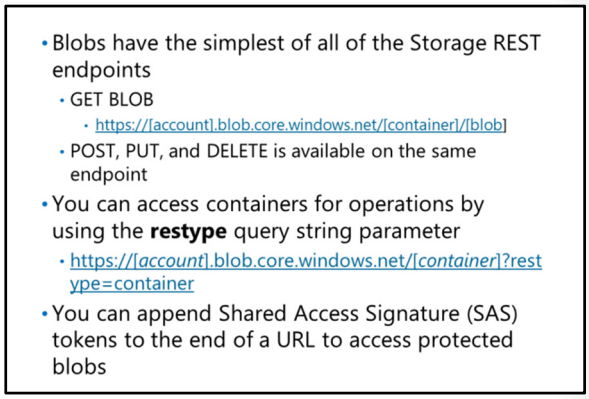
Page Blobs

Page blobs are blobs that are composed of individual pages that are optimized for random read-write operations. These individual pages are 512 bytes. When creating a page blob, you must specify the maximum size for the blob. If you need more space for your file, you must create a new page blob. To update the page blob, you must write one or more 512-byte pages. When writing the pages, an offset is required along with a range that within the 512-byte page boundary. When updating a page blob, you can overwrite up to 4 megabytes (MB) of pages. Page blob write operations are immediately committed and overwrite the in-place pages. A page blob is typically used for virtual hard disks and can be up to 1 terabyte (TB).

REST API for Storage Blobs

An extensive REST API for Storage is already available. This API allows you to manage accounts in a RESTful manner. For blobs, this API has been extended to ensure that it is easy to access a blob by using a simple URL.

You can access blobs by using the **GET**, **PUT**, **POST**, or **DELETE** HTTP methods. You can access blobs by using the following URL format:



https://[*account name*].blob.core.windows.net/[*container name*]/[*blob name*]

You can provide additional query string parameters for additional metadata and functionality such as:

- **Snapshot**. Retrieves a snapshot copy of a blob at the specified datetime value
- **Restype**. Can be used directly with the container to access the container metadata or enumerate blobs in a container

In addition to the traditional Create, Read, Update, Delete (CRUD) functionality, the metadata and properties of a blob can be accessed by using the REST API. Finally, access policies (ACL) can be managed by using the same REST API.

Lesson 2
Controlling Access to Storage Blobs and Containers

Sometimes files need to be secured so that they cannot be accessed by anonymous entities. Storage blobs contain a mechanism for securing blobs and granting temporary access by using query string tokens.

This lesson introduces the concept of blob permissions and SAS tokens.

Lesson Objectives

After completing this lesson, you will be able to:

- Implement security on a container or a blob.
- Generate a SAS token.
- Create a shared access policy that can be used to generate SAS tokens.

Container Permissions

Typically, only the owner of a storage account can access resources within that account. If your service or application needs to make these resources available to other clients, you have various options available. First, you can make the public access key generally available. This is not typically recommended as this key gives individuals full access to your entire storage account and its management operations. Another, more common option is to manage access for the entire container.

- There are three levels of container access that are available
  - Full public read access
    - Enumerate container blobs
    - Read individual blobs
    - Cannot enumerate containers
  - Public read access for blobs only
    - Read individual blobs
  - No public read access
    - No access to blobs, containers, or enumerating contents

The Public Read Access container setting:

- Container Public Read Access setting

	Anonymous			Key
	Enumerate Containers	Enumerate Container Blobs	Read Blob	Read Blob
Container		✓	✓	✓
Blob			✓	✓
Off				✓

FIGURE 7.2: PUBLIC READ ACCESS

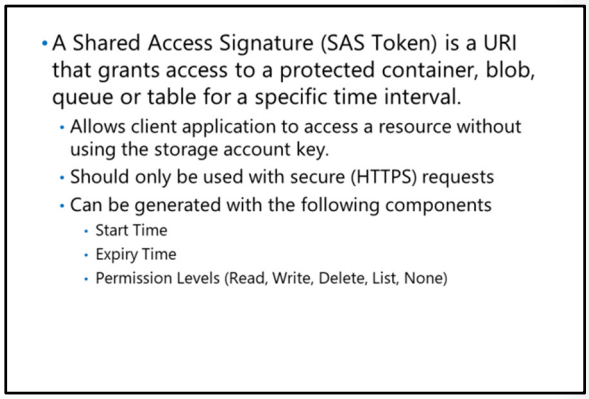
The Public Read Access property controls what data is available anonymously for your container. You can select the following values for the Public Read Access setting:

- **Container**. Blobs in a container can be enumerated. The container metadata is also accessible. Individual blobs within this container and their properties can also be accessed with this setting.

- **Blob**. Only individual blobs and their properties in this container can be accessed. Blobs are not allowed to be enumerated.
- **Off**. With this setting, enumeration of blobs is not allowed. Individual blobs and their properties are also not accessible. You must use your access keys to access any data about this container or it's blobs.

Shared Access Signatures

A Shared Access Signature (SAS) token is a set of URL query string parameters and values that can grant access to a restricted container, blob, queue or table. The SAS token can either contain details about the level of access or a reference to a policy that contains those details. Some of the operations that you can be exposed using SAS includes:



- Reading or modifying page/block blob content.
- Reading or modifying blob properties and metadata.
- Creating, leasing or removing a blob snapshot.
- Creating and managing queue messages.
- Reviewing queue metadata including the queue length.
- Querying, creating, managing and updating table entities.

In order to apply the SAS parameters, the query string parameters are appended to the end of the base URL that is typically used to access the Storage resources in an anonymous manner. The query parameters in this URL now contain the details necessary to grant temporary access to a restricted resource. The query parameters include data such as:

- Valid time interval for SAS token.
- SAS token start time.
- Permissions granted to resource (read/write/read+write).
- Signature validating the SAS token.

The SAS token can also reference an existing Stored Access Policy that provides additional control over the SAS tokens. The policy can be reused to revoke existing signatures or to standardize access options for each generated signature.

When using a SAS token, you should always use HTTPS in your requests. The query parameters in your URL contain data that can possibly be used in a malicious manner with your storage account. For example, if a SAS token has an expiration time of 1 day, and your request is intercepted. The interceptor can re-use the SAS token in their requests. If the token could possibly grant access to unintended blobs that are in the same container.

The following code example creates an access policy on a container, and then generates a shared access signature for the container.

Creating a Shared Access Signature

```
BlobContainerPermissions blobPermissions = new BlobContainerPermissions();
blobPermissions.SharedAccessPolicies.Add("mypolicy", new SharedAccessBlobPolicy()
{
   SharedAccessExpiryTime = DateTime.UtcNow.AddHours(10),
   Permissions = SharedAccessBlobPermissions.Write |
      SharedAccessBlobPermissions.Read
});
blobPermissions.PublicAccess = BlobContainerPublicAccessType.Off;
container.SetPermissions(blobPermissions);

string sasToken =
   container.GetSharedAccessSignature(new SharedAccessBlobPolicy(), "mypolicy");
```

You can use a SAS token by appending it to the end of the REST URL for this blob resource. The SAS token is traditionally a simple query string value.

Stored Access Policies

A Stored Access Policy can provide additional functionality to your Shared Access Signatures. Creating policies allows you to group your SAS tokens and control signatures generated by this policy even after they are generated. In the policy, you can define settings such as:

- Start time
- Expiry time
- Signature permissions

- Stored access policies allow you to have granular control over a set of shared access signatures
  - The signature lifetime and permissions are stored in the policy as opposed to the URL
  - You can modify the properties in the policy and the changes are automatically propagated to all signatures generated from the policy
  - You can also invalidated all signatures generated from a policy
- A Container, Queue or Table can have up to 5 Stored Access Policies

Signatures generated using a policy can be revoked by your applications even after the signatures have been distributed to client applications or devices. Another advantage of using policies is that the signature's permissions and lifetime are no longer specified in the URL of the request. If you need to change these parameters for one or more existing signatures, you can simply update the policy. This avoids reissuing signatures. In the case of a compromise, you can easily revoke all signatures associated with the policy.

Best Practice: It is considered a best practice to use a policy instead of individual signatures for applications that generates multiple SAS tokens for client devices. The stored access policy allows you to modify or remove the existing signatures after they are issued. If a policy is not used, the recommendation is to at least minimize the lifetime of your signature to minimize risk to your existing storage account resources.

To use a stored access policy, you must first create the policy and associated it with your storage account. The policy must contain a unique identifier which is a string with a maximum length of 64 characters. After the policy is created, you specify the unique identifier of the policy when creating new signatures. These signatures are now associated with the policy and will specify the policy in their query string parameters. To update a policy or revoke the signatures for a policy, you simply replace the policy in the list of policies for your storage account with a new policy using the same unique identifier.

Generating Shared Access Signatures from Policies

By using stored access policies and shared access signatures, you can implement the Valet Key pattern for Storage blob access.

Typically, the web service accesses blob resources directly and downloads the media, on behalf of the user, before sending it to the user.

Proxy web service returning data from storage:

```
// Get a reference to the container
var container = blobClient.GetContainerReference("files");
container.CreateIfNotExists();

// Create blob container permissions
var blobPermissions = new BlobContainerPermissions();
blobPermissions.SharedAccessPolicies.Add("mypolicy", new
     SharedAccessBlobPolicy() {
          SharedAccessExpiryTime =
  DateTime.UtcNow.AddHours(10),
          Permissions = SharedAccessBlobPermissions.Read
     }
);
blobPermissions.PublicAccess =
BlobContainerPublicAccessType.Off;
```

FIGURE 7.3: STORAGE PROXY

By using the Valet Key pattern, the web service can receive a request from the user, validate the user, and then return the credentials necessary for the user to access the resource directly. The web service accomplishes this by using a stored access policy and generating a shared access signature. The signature is then appended to the end of the blob URL and the entire URL along with the signature is returned to the user. The user can then use the expanded URL to access the blob resource directly from Azure storage in a controlled manner.

A provider web service returning the URI and token from Storage and allowing the client to download data directly:

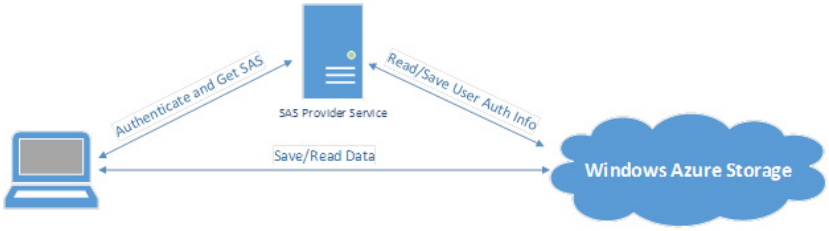


FIGURE 7.4: SEPARATING AUTHENTICATION AND ACCESS USING AZURE STORAGE

Reference Link: https://docs.microsoft.com/azure/storage/common/storage-dotnet-shared-access-signature-part-1

Lesson 3
Configuring Azure Storage Accounts

Azure storage accounts can have additional features enabled or configured to extend the base functionality.

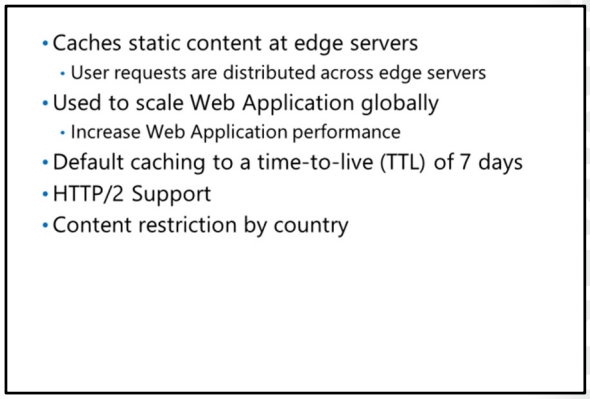
This lesson introduces the CDN and cross-origin resource sharing (CORS) features for a storage account.

Lesson Objectives

After completing this lesson, you will be able to:

- Enable and use CDN to create low-latency replicas of content.
- Enable and use CORS to enable cross-domain client script support.

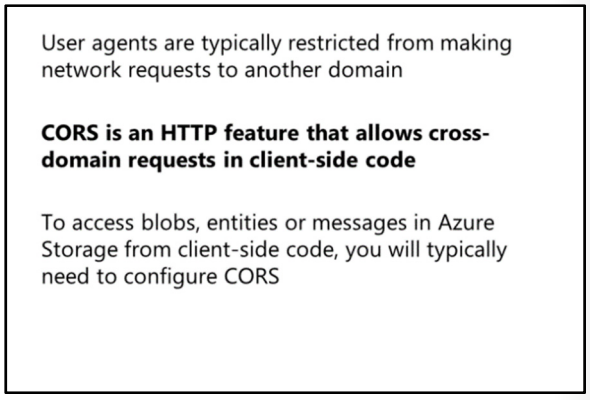
Azure Content Delivery Network (CDN)



A Content Delivery Network (CDN) offers content and multimedia to client devices in a high-efficiency manner by placing the content "closer" to the end users. For example, the Azure CDN has nodes in the United States, Europe and Australia along with other locations. Users download content from the closest node to their current location so that every user has the fastest possible download of your content. Typically, if a user lives far from the source of your content, they will have a worse download experience for the content than users who live closer to the content. The Azure CDN replicates content across multiple geo-distributed nodes so that more users see better performance and an enhanced user experience regardless of their location.

The Azure CDN service replicates content in your Storage account across nodes in various regions. The CDN service is billed separately from your storage account. To use the CDN service, you create a CDN endpoint within your storage account. The storage account provides origin data and caches them to CDN nodes across various regions. The CDN service is also compatible with Azure Cloud Services. Content in the CDN remains there until it expires. This expiration is determined by using the configurable time-to-live value. All publicly available content in your storage account or hosted service is cached to your CDN. If any of the content is modified, the changes not be available in the CDN until either the CDN refreshes its content through manual intervention or the cached CDN content's time-to-live period of time expires.

Cross-Origin Resource Sharing



If you request a storage resource from a web application under a separate domain, you are typically making a Cross-Origin Resource Sharing request. This is a special type of that allows resources to be accessed outside of their originating domain. A CORS request is normally made in two parts. First a pre-flight request is made by using the HTTP method OPTIONS. This request determines if your host domain has permission to perform the cross-origin request. The response returns the allowed origins and methods in its header values. The headers of the response will let the browser know if the request is allowed. After this is verified, a second request is sent using the original HTTP method (GET, PUT, POST, and DELETE). CORS is supported for Blob, Table, and Queue services in Azure. Other Azure services such as Search and API Management also have similar CORS functionality.

Reference Link: https://www.w3.org/TR/cors/

By default, CORS is not enabled for your storage account. For each Storage service, Blob, Table, and Queue, you must individually configure CORS rules either using an automation script or either of the portals. When you configure CORS, you will specify a range of permitted origins, methods, and headers.

The following is a sample of a single CORS rule:

CORS Rule

```
<Cors>
      <CorsRule>
            <AllowedOrigins>http://www.contoso.com,
http://www.fabrikam.com</AllowedOrigins>
            <AllowedMethods>PUT,GET</AllowedMethods>
            <AllowedHeaders>x-ms-meta-data*,x-ms-meta-target*,x-ms-meta-
abc</AllowedHeaders>
            <ExposedHeaders>x-ms-meta-*</ExposedHeaders>
            <MaxAgeInSeconds>200</MaxAgeInSeconds>
    </CorsRule>
<Cors>
```

The CORS rule is an XML document with specific expected elements. The most common elements are described below:

- **AllowedOrigins**. This is a list of origin domains that are allowed to make a cross-origin request against your storage service. The origin is determined by validating the Origin header from the client request. A wildcard character can optionally be used here to indicate that all origins are allowed. In the example above, by using CORS, the domains http://www.contoso.com and http://www.fabrikam.com can make requests against the service.
- **AllowedMethods**. This is a list of HTTP methods that are allowed in your cross-origin requests. Pre-flight requests using the OPTIONS method are implicitly allowed and do not need to be specified in this document. In the example above, only PUT and GET requests are permitted.

- **MaxAgeInSeconds**. This specifies how long a browser can cache the response to the pre-flight OPTIONS request. This is useful in scenarios where you need to regularly update the CORS rules for your storage account. In the example above, a browser must issue a new pre-flight request if the cached OPTIONS response is older than 200 seconds.

Lesson 4
Azure Files

File is a service that provides an SMB file share that can be used to share files among multiple virtual machines.

This lesson describes the File service and the file share logical unit.

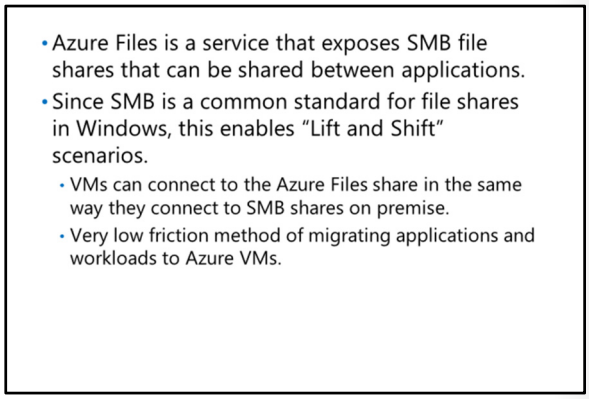
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the File service.
- Describe the relationship between a file share and an SMB share.

Azure Files Overview

Many existing or inherited applications rely on specific file structure on your systems. Although the guidance is to design new cloud applications to not rely strongly on local storage, the reality is that many applications already have strong dependencies on local storage. In many business contexts, it is ideal to deploy the application quickly using a "lift and shift" process rather than wait for the application to be re-engineered prior to deploying it to cloud infrastructure.

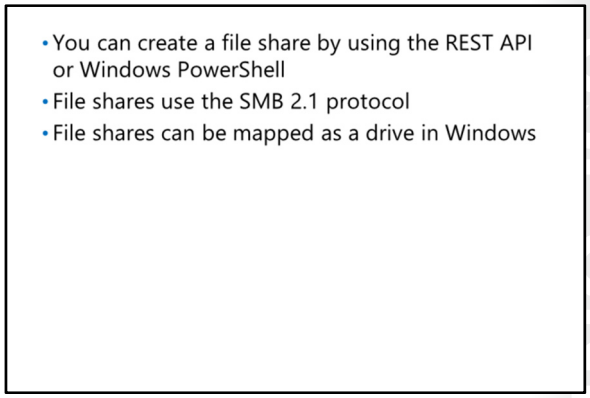


Azure Files is a component of the Storage service. Azure Files exposes a SMB file share that can be mounted in Azure service instance. This file share can be used for various tasks such as sharing data between multiple virtual machines or migrating data from local infrastructure to an Azure virtual machine. Azure Files is a popular option for migrating existing applications and their data from on premise to a collection of virtual machines in Azure.

Applications running in Azure virtual machines or Cloud Services can mount the new file share using the SMB 2.1 protocol. This is supported in both Linux and Windows operating systems that support the protocol. The process to mount the share is the same as mounting any other SMB file share. Multiple application components can mound the shares and use them to share data simultaneously.

Azure Files exposes a REST API in a manner similar to other Azure services. On premise applications can use this REST API to access data in the share or to add data to the share. Using this API, applications can operate in a hybrid manner with components hosted both in Azure and on premise.

File Shares



Because a File storage share is a standard SMB 2.1 file share, applications running in Azure can access data in the share through the file system I/O APIs. Developers can therefore use their existing code and skills to migrate the applications. IT Professionals (IT pros) can use Windows PowerShell cmdlets to create, mount, and manage File storage shares as part of the administration of Azure applications.

Distributed applications can also use File storage to store and share useful application data, development tools, and testing tools. For example, an application might store configuration files and diagnostic data such as logs, metrics, and crash dumps in a File storage share so that they are available to multiple virtual machines or roles. Developers and administrators can store utilities that they need to build or manage an application in a File storage share that is available to all components, rather than installing them on every virtual machine or role instance.

Lab: Storing Generated Documents in Azure Storage Blobs

Scenario

You need a place to store the Word documents that are generated by the Contoso Events application. You decide to store the generated Word documents in blobs. You also decide to create a protected container so that the Word documents are not accessed by anonymous users. Finally, you want to create the logic to generate SAS tokens for temporary access to one of the Word documents.

Objectives

After you complete this lab, you will be able to:

- Create a container by using the Azure Management Portal.
- Add blobs to the container by using the Azure Storage SDK.
- Download blobs from the container by using MemoryStream objects.
- Generate SAS tokens for accessing a blob.

Lab Setup

Estimated Time: 60 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Implementing Azure Storage Blobs

Exercise 2: Populating the Container with Files and Media

Exercise 3: Retrieving Files and Media from the Container

Exercise 4: Specifying Permissions for the Container

Question: Your server application needs to generate a SAS for the client to download a subset of blobs in a single container. Should you give the signature read access to the container or read access to the individual blobs?

Module Review and Takeaways

In this module, you used Storage blobs to store files for your cloud application. You also learned how to use the Azure Storage SDK to manage blobs, containers, and their permissions. Finally, you explored the mechanisms that are used to generate SAS tokens for a container.

Review Question

Question: You have a web application that allows users to download a large blob that is protected. Your server-side logic generates a SAS token to retrieve the protected blob. Should your server download the blob and stream it to the web client or should your server provide the blob URL with the appended SAS token to the web client?

Module 8

Designing a Communication Strategy by Using Queues and Service Bus

Contents:

Module Overview

With web applications presenting content and worker roles processing the logic, there needs to be a mechanism that facilitates the communication between these different entities. Microsoft Azure provides two queuing mechanisms that you can use for this purpose. Lesson 1, "Azure Storage Queues," introduces the queue mechanism that is available in Azure storage accounts. Lesson 2, "Azure Service Bus," introduces the Service Bus offering in Azure. Lesson 3, "Azure Service Bus Queues," describes the queuing mechanism that is available in Service Bus and how it differs from Azure Storage queues. Lesson 4, "Azure Service Bus Relay," describes the relay mechanism available to connect client devices to WCF services. Lesson 5, "Azure Service Bus Notification Hubs," introduces the Notification Hubs service and infrastructure useful for pushing notifications to mobile devices.

Objectives

After completing this module, you will be able to:

- Describe Storage Queues service.
- Describe Service Bus.
- Describe Service Bus Queues service.
- Describe Service Bus Relay.
- Describe Notification Hubs service.

Lesson 1
Azure Storage Queues

Storage queues provide a consistent and reliable way to store messages that can be consumed by multiple workers.

This lesson introduces the Queue service in Storage and describes some of its characteristics.

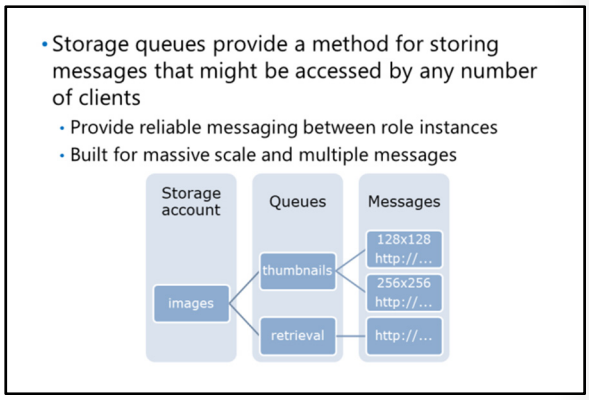
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Storage queues.
- Describe the characteristics of Queue messages.
- View messages in Storage queues.

Storage Queues Overview

Queue is a service for storing large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. A single queue message can be up to 64 kilobytes (KB) in size, and a queue can contain millions of messages, up to the total capacity limit of a storage account. A storage account can contain up to 200 terabytes (TB) of blob, queue, and table data.



Common Uses of Storage Queues

- Creating a backlog of work to process asynchronously
- Passing messages from an Azure web role to an Azure worker role

The Queue service contains the following components:

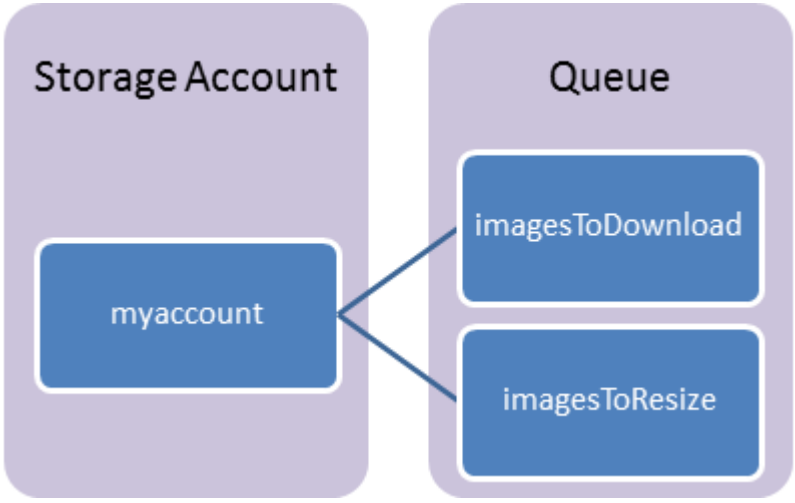


FIGURE 8.1: QUEUE COMPONENTS

URL format: You can address queues by using the following URL format:

http://<storage account>.queue.core.windows.net/<queue>

The following URL addresses one of the queues in the diagram:

http://myaccount.queue.core.windows.net/imagesToDownload

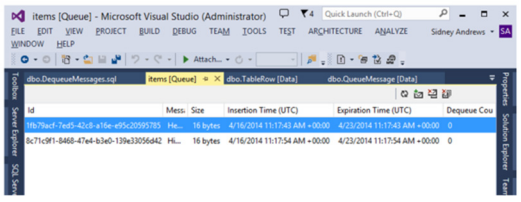
Storage Account: All access to Storage is done through a storage account.

- **Queue**: A queue contains a set of messages. All messages must be in a queue.
- **Message**: A message, in any format, of up to 64KB.

Viewing Queue Storage Data

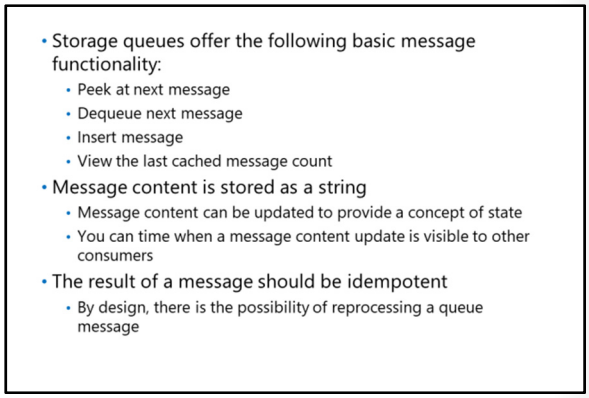
Starting with Microsoft Visual Studio 2015, the Server Explorer pane gives you the ability to manage and view Azure service instances directly within the Integrated Development Environment (IDE). You can view the messages within a queue by using the Server Explorer.

Visual Studio 2015 Server Explorer provides a view for Storage queue items in the emulator or in a live Azure storage account.



Storage Queue Messages

The content in a queue message is stored as a string. You can store complex objects in a queue by serializing them as a string.



Common Queue Message Actions

- Add Messages
  - Messages can be added to the queue
- Get Message or Messages
  - Retrieves the next message or messages from the queue and they will be invisible to other clients for a specified timeout period
- Peek Message or Messages
  - Retrieves the next message or messages from the queue while keeping the message or messages visible to other clients
- Update Message
  - Updates the content or visibility timeout of a specified message
- Delete Message
  - After a message is processed, you can delete the message from the queue so that it won't be processed again.

Timeouts and Idempotent Processing

Messages should be designed such that they can be processed multiple times without causing side effects. This occurs because a message might become visible to other queue clients if the visibility timeout elapses and the message that was retrieved is not deleted. Common scenarios where this might occur include worker role failure or transient service errors.

Updating a Queue Message

Messages that exist in a queue are not static. You can change the contents of any message in a queue or modify the visibility timeout for an existing queue message. For example, if your application has a predicted failure, you can save the current status of a message that is being processed and either extend the visibility timeout or have the message handled by another worker instance. In a typical implementation, metadata about the amount of retried attempts for a queue message is stored in the message properties. This can help prevent an endless loop of processing a message that causes a particular application failure. Updating queue messages can also be used as a strategy for creating a multistep workflow on queue messages where they are handled by various different worker instances.

Updating message content and making it visible immediately.

Update Message

```
CloudQueueMessage message = queue.GetMessage();
message.SetMessageContent("Updated contents.");
queue.UpdateMessage(message,
    TimeSpan.FromSeconds(0.0),  // Make it visible immediately.
    MessageUpdateFields.Content | MessageUpdateFields.Visibility);
```

Lesson 2
Azure Service Bus

Service Bus is a fully managed messaging platform in Azure. Components of your application can leverage Service Bus to share messages in a disconnected manner.

This lesson describes the Service Bus service and its features.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe Service Bus.
- Describe the features of Service Bus.
- Create a Service Bus namespace.

Service Bus Overview

Different situations require different styles of communication. Sometimes, letting applications send and receive messages through a simple queue is the best solution. In other situations, where an ordinary queue isn't enough, a queue with a publish and subscribe mechanism is better. And in some cases, all that's really needed is a connection between applications—queues aren't always required for intermodule communication. Service Bus provides all three options, letting your applications interact in several different ways.

- Service Bus is a managed messaging infrastructure
  - Massive in scale and completely managed
  - Allows you to scale out your applications and consumers knowing that the messaging platform will scale out with your application
- Allows decoupled components to communicate asynchronously and synchrounously

Service Bus is a multitenant cloud service, which means that the service is shared by multiple users. Each user, such as an application developer, creates a namespace, and then defines the communication mechanisms he or she needs within that namespace.

Service Bus provides a multitenant service for connecting applications through the cloud.

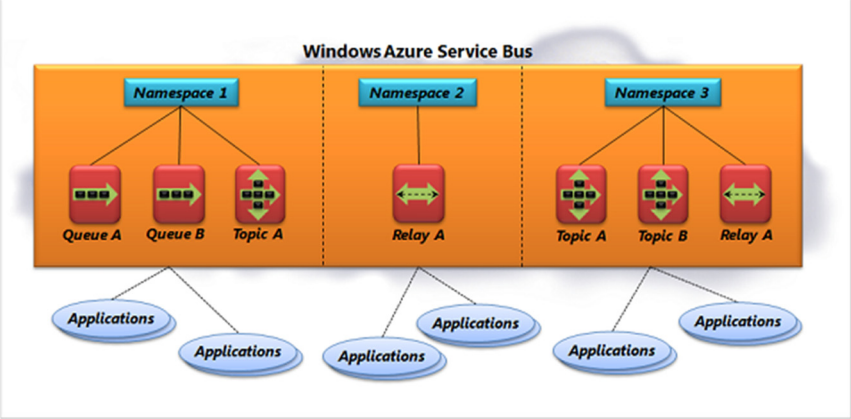
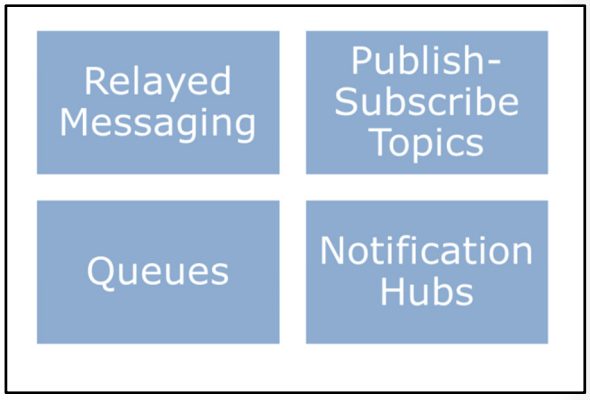


FIGURE 8.2: SERVICE BUS ARCHITECTURE

Service Bus Features

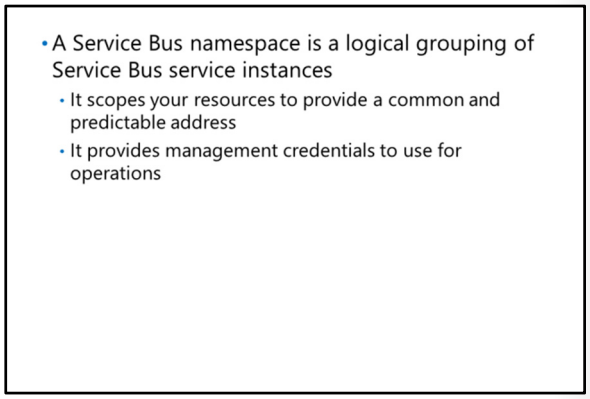
Service Bus is composed of four communication services. You can create one or more instances of each service to connect toy our applications. These services include:



- **Queues**. They act as an intermediary layer between your application components and store messages that other application components can receive.
- **Topics**. They are a one-directional communication mechanism that allow client applications or devices to subscribe to a topic. A separate application or device can publish messages to topics for consumption by the client applications.
- **Relays**. They are a bidirectional proxy for communication with a Windows Communication Foundation (WCF) service. Client applications can bind directly to the relay endpoint and the relay infrastructure handles routing messages to the appropriate WCF service endpoint.
- **Notification Hubs**. It is a managed, brokered system for distributing messages from server applications to client devices across various platforms by using local notifications.

Namespaces

Namespaces serve as a basic logical grouping of Service Bus service instances.



When you create a queue, topic, or relay, you give it a name. The instance name is then combined the name of your namespace to create a unique identifier for the object. Applications can provide this name to Service Bus, and then use that queue, topic, or relay to communicate with one another.

Service Bus namespaces can also contain management credentials, or shared keys, that your client applications can use to connect to Service Bus.

Lesson 3
Azure Service Bus Queues

Service Bus provides the queue functionality that you can use to marshal messages from reporting applications to consuming applications. A Service Bus queue is different from a Storage queue.

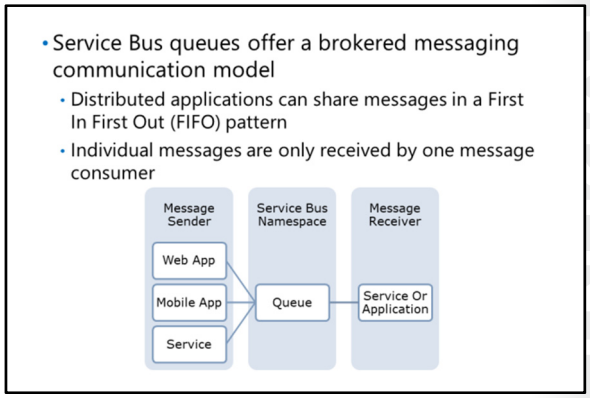
This lesson describes Service Bus queues and the difference between Service Bus queues and Storage queues.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe Service Bus queues.
- Describe the differences between Service Bus queues and Storage queues.

Service Bus Queues Overview



Service Bus Queue is a brokered messaging system, which is similar to the Queue service Azure Storage. By using these queues, application modules that are distributed do not need to communicate directly with each other. These application modules can instead communicate by using the queues. This ensures that there is a separation between message generators and message processors. This separation provides the flexibility of having one or more application component instances that are generating messages and one or more application component instances that are processing the same messages. If an instance encounters an irrecoverable exceptional condition, other instances can continue processing the messages. If the workload for the entire application is increased, new instances can be created to handle the load. These scenarios are common and critical when developing and designing cloud applications.

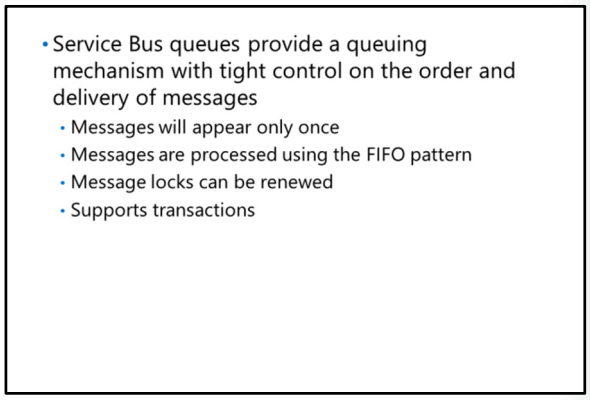
Service Bus queues do implement a familiar first in, first out (FIFO) message delivery strategy. Service Bus queues can also guarantee that a message is received and processed both at least and at most once by the message consumers.

Service Bus queues are a general-purpose technology that you can use in a wide variety of scenarios:

- Communication between web and worker roles in a multitier Azure application
- Communication between on-premises apps and Azure hosted apps in a hybrid solution
- Communication between components of a distributed on-premises application that is running in different organizations or departments of an organization

By using queues, you can scale out your applications better, and provide more resiliency to your architecture.

Queue Message Delivery



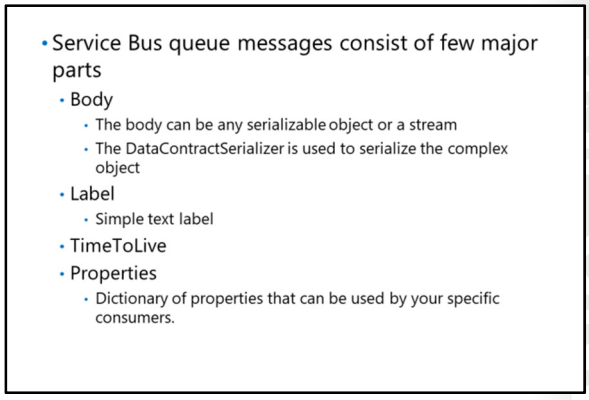
Messages in a Service Bus queue are delivered and handled in a common, predictable pattern. First, a message generator creates a new message and adds the message to a queue. A receiver can process this message at a later time. Many receivers can process messages in the same queue, but they do not share these messages in a multicast scenario. Messages are typically processed by only one receiver unless there is an exception condition.

When a receiver reads a message, it has two options to handle the message. The receiver can first choose the **ReceiveAndDelete** method. This method removes the message from the queue and deletes it after it is read. The risk with using this method is that the message can be lost if there is an error with the receiver's application. Alternatively, the receiver can choose to use the **PeekLock** method. This method still removes the message from the queue but it does not delete the message. The message is instead locked and is flagged as not visible to other receiver instances. This lock can be handled in three ways:

- If the message is successfully processed, the **Complete** method is used to instruct the queue to delete the message immediately.
- If the message has failed at some point in processing but the application can handle this failure without crashing, the **Abandon** method is used to indicate that the lock should be removed from the message. The message will now be visible to other receivers that wish to process the message.
- If the configured time-to-live value for the queue message expires, the queue assumes that the receiver is in a faulted state. The queue removes the lock from the message and makes it available to other receivers.

After a receiver application component reads the message, it is the responsibility of the receiver to call the Complete method after the message is successfully processed. If this does not occur, an infinite loop can happen where the same message is processed infinitely because of the time-to-live period expiring for the message. Service Bus Queue messages have a unique ID that your client application can use to determine if a message is processed by more than one receiver.

Characteristics of Service Bus Queue Messages



If you use the .NET libraries for Azure, you must use the **BrokeredMessage** class to represent the Service Bus Queue messages. The **BrokeredMessage** class includes standard properties that represent metadata that is pertaining to the individual message. These properties include:

- **Label**. This property that can be used by your applications to provide details and simple labels for messages.
- **TimeToLive**. You can use this property to indicate the duration that a message should be persisted in the message store.
- **MessageId**. You can use this property to provide a unique identifier for the message.

The message object also includes a dictionary property named **Properties**. This **IDictionary<string, object>** typed property can contain any custom properties that you want to define for your application. These properties are included in the object in addition to the actual body of the message. The message body is a Common Language Runtime (CLR) object that is serialized by using **DataContractSerializer**. After you create a new **BrokeredMessage** instance, you can add the message body by using the **GetBody<T>** method of the **BrokeredMessage** class. Because the body is serialized, this can be any complex object that you want to transmit from your sender application to receiver applications. Additional metadata about this message is typically stored in the **Properties** property of the **BrokeredMessage** class.

This example demonstrates how to send five test messages to a **QueueClient** instance.

New BrokeredMessage

```
BrokeredMessage message = new BrokeredMessage("Test message ");

message.Properties["TestProperty"] = "TestValue";
message.Properties["Message number"] = 12;

Client.Send(message);
```

Each message has two parts, a set of properties, representing a key/value pair, and a binary message body. How they are used depends on what an application is trying to do. For example, an application sending a message about a recent sale might include the properties *Seller="Ava"* and *Amount=10000*. The message body might contain a scanned image of the sale's signed contract, or, if there isn't one, the message body might be empty.

You can retrieve messages by using similar properties from the **BrokeredMessage** class.

This example demonstrates how messages can be received and processed by using the default **PeekLock** mode.

Retrieving messages

```
while (true)
{
    BrokeredMessage message = Client.Receive();

    if (message != null)
    {
        try
        {
            Console.WriteLine("Body: " + message.GetBody<string>());
            Console.WriteLine("Test Property: " +  message.Properties["TestProperty"]);

            message.Complete();
        }
        catch (Exception)
        {
            message.Abandon();
        }
    }
}
```

Service Bus Queues vs. Storage Queues

Azure supports two types of queue mechanisms, Storage queues and Service Bus queues.

- **Storage queues**. These are part of the Azure storage infrastructure. They have a simple REST-based Get/Put/Peek interface. They provide reliable, persistent messaging within and between services.

Storage Queues	Service Bus Queues
• Arbitrary ordering	• FIFO guaranteed ordering
• Delivery at least once, possibly multiple times	• Delivery at least once and at most once
• 30 second default locks can be extended to 7 days	• 60 second default locks can be renewed
• Supports in-place updates of the message content	• Messages are finalized once consumed
• Can integrate with WF through a custom activity	• Native integration with WCF and WF

- **Service Bus queues**. They are part of a broader Azure messaging infrastructure that supports queuing and publish/subscribe, web service remoting, and integration patterns.

Although both queuing technologies exist concurrently, Storage queues are introduced first, as a dedicated queue storage mechanism built on top of Azure Storage services. Service Bus queues are built on top of the broader brokered messaging infrastructure that is designed to integrate applications or application components, which might span multiple communication protocols, data contracts, trust domains, and network environments.

Differences Between Storage Queues and Service Bus Queues

Comparison Criteria	Storage Queues	Service Bus Queues
Ordering guarantee	No	Yes first in first out (FIFO)
Delivery guarantee	At-Least-Once	At-Least-Once At-Most-Once
Transaction support	No	Yes
Receive mode	Peek and Lease	Peek and Lock Receive and Delete
Exclusive access mode	Lease-based	Lock-based
Lease/Lock granularity	Message level	Queue level
Batched receive	Yes	Yes
Batched send	No	Yes

- **Ordering guarantee**. Service Bus queues guarantee that messages are processed in a FIFO order. Storage queues do not guarantee this.
- **Delivery guarantee**. Service Bus and Storage queues guarantee that messages are delivered at least once. Service Bus queues can also guarantee that a message is delivered *only* once.
- **Lease granularity**. Storage queues can set the lock/lease length to a different value for each individual message. Service Bus queues can set this value only for all the messages in a queue instance.
- **Batches**. Both Storage and Service Bus queues can receive messages in a batch. Only Service Bus queues support batch creation of messages.

Lesson 4
Azure Service Bus Relay

Service Bus relays provide a flexible way to connect WCF services and clients without having to redesign the network architecture in your organization. When you connect the WCF service to a relay in an outgoing manner, you clients will need to connect only to the Service Bus endpoint in Azure to communicate with your WCF service.

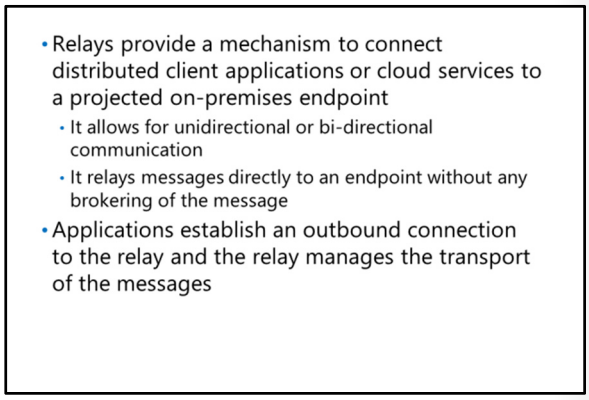
This lesson describes the benefits and architecture of Service Bus relays.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the benefits of using a Service Bus relay.
- Detail the architecture of Service Bus relays.

Service Bus Relay Overview



Service Bus includes a relay component that can connect existing services to new client applications without exposing the true location or address of the service. Supported services include Simple Object Access Protocol (SOAP), Web Services standards (WS*), and Representational State Transfer (REST). Some of the advantages of using a relay include:

- WCF Services that need to communicate directly with external client applications or devices (For example, mobile devices) typically need to be placed in a special subnet or a virtual network with a unique NAT or firewall configuration. The address for the WCF endpoint will need to be publicly addressable in order for client devices to connect. In some enterprises, this can be considered dangerous or unacceptable. With Service Bus Relay, the WCF service makes an outbound connection to the relay and bypasses many of the complex network configurations that are necessary for inbound connections.
- Although mobile applications are deployed and updated regularly, end users might not update their applications as regularly as you want them to. If your WCF service needs to be migrated to a new network or moved to a new IP address, this can cause a lapse of connectivity for your mobile applications. Using Service Bus Relay, your mobile applications address a publicly accessible and permanent uniform resource identifier (URI). You are then free to make changes and migrate your WCF service within your organization's infrastructure. The new service instance or location simply needs to connect to the relay for client devices to access it. This enables more mobility for services that are connected to the applications that are already deployed.

Service Bus Relay also supports direct peer-to-peer communication. This is negotiated if the relay determines that the connecting application can easily and directly address the WCF service.

The Service Bus Relay service enables you to build hybrid applications that run in an Azure data center and in your own on-premises enterprise environment.

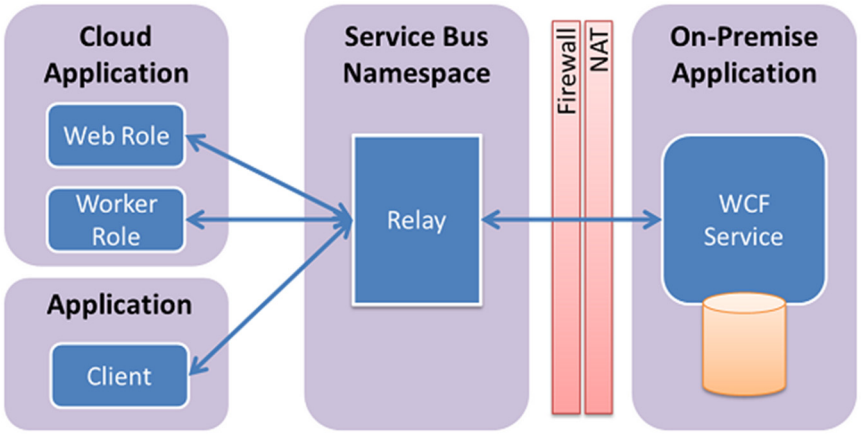


FIGURE 8.3: SERVICE BUS RELAY

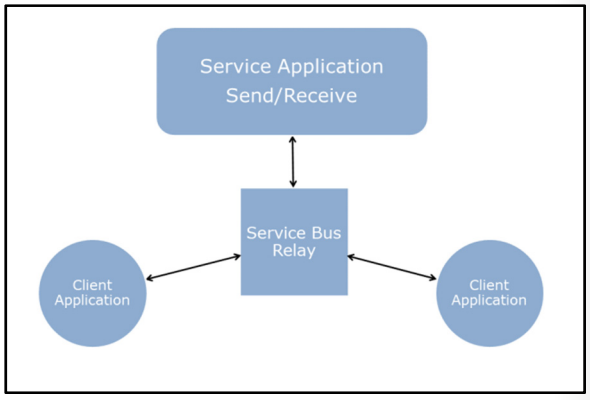
When you use Service Bus Relay, you must use a set of WCF bindings that are similar to the bindings that ship normally with WCF. These bindings include a relay prefix. They implement new binding elements that create a channel to your Service Bus instance in Azure. Bindings are included for one-way messages, request/response messages, and distributed event messages. The event messages are unique to Service Bus Relay and are used to enable a publish/subscribe scenario where your client applications can send a message and have that message distributed to multiple WCF service instances. For example, event messaging can be used to distribute a message to a receiver WCF service for processing and auditing a WCF service.

Because Service Bus Relay utilizes bindings that are very similar to the traditional WCF bindings, many applications do not require large changes to use Service Bus Relay. The client applications typically require a new binding configuration that specifies the relay binding. After the client communicates with the relay, the relay service handles the targeting of messages to the appropriate WCF services. The client application does not need to know anything about the service's address or actual location. The service applications do not need to expose inbound ports on their firewall to receive these messages.

Benefits of Service Bus Relay

- **Flexible NAT configuration**. Because the relay makes an outbound connection, you can use it in complex network environments where you would normally require changing your firewall for inbound connections.
- **WCF**. Relay is used with WCF services. WCF is a mature, stable messaging framework that supports both SOAP and REST messaging. WCF also has a mature ecosystem of custom bindings, behaviors, and components. SOAP and REST is also already widely supported by many existing services or client devices.

Service Bus Relay Architecture



The Service Bus relay allows you to host WCF services within your existing enterprise environment. After you host, you can delegate the task of listening to incoming sessions and requests to the Service Bus service running within Azure. These messages are then routed to the appropriate WCF service. This enables you to expose these services to the application code that is running in Azure, or to mobile workers or extranet partner environments. Service Bus allows you to securely control who can access these services at a fine-grain level. It provides a powerful and secure way to expose application functionality and data from your existing enterprise solutions and take advantage of it from the cloud.

Incoming Relay Requests

When a client sends a request to Service Bus, the Azure load balancer routes it to any of the gateway nodes. If the request is a listening request, the gateway node creates a new relay. If the request is a connection request to a specific relay, the gateway node forwards the connection request to the gateway node that owns the relay. The gateway node that owns the relay sends a rendezvous request to the listening client, asking the listener to create a temporary channel to the gateway node that received the connection request.

Management Credentials

- Service Bus uses a Shared Access Signature (SAS) to authenticate access to the messaging entities within the namespace
  - This replaces the ACS functionality previously available
- You can also use a simple web token (SWT) or SAML token from a provider

Applications can authenticate to Service Bus by using Shared Access Signature (SAS) authentication. Previously, Azure Active Directory Access Control Service (ACS) was used to provide an access key.

SAS authentication enables you to grant a user granular access to Service Bus resources with specific rights. SAS authentication in Service Bus involves the configuration of a cryptographic key with associated rights on a Service Bus resource. Clients can then gain access to that resource by presenting a SAS token, which consists of the resource URI being accessed and an expiry signed with the configured key.

You can configure keys for SAS on a Service Bus namespace. The key applies to all messaging entities in that namespace. You can also configure keys on Service Bus queues, topics, and notification hubs. Support for Service Bus relays will be added in the near future.

To use SAS, you can configure a **SharedAccessAuthorizationRule** object on a namespace, queue, topic, or notification hub that consists of the following:

- KeyName that identifies the rule
- PrimaryKey, a cryptographic key, which is used to sign or validate SAS tokens

- SecondaryKey, a cryptographic key, which is used to sign or validate SAS tokens
- Rights that represent the collection of Listen, Send, or Manage rights granted

Authorization rules configured at the namespace level can grant access to all the entities in a namespace for clients with tokens signed by using the corresponding key. You can configure up to 12 such authorization rules on a Service Bus namespace, queue, topic, or notification hub. By default, a **SharedAccessAuthorizationRule** object with all rights is configured for every namespace when it is first provisioned.

Lesson 5
Azure Service Bus Notification Hubs

Notification Hubs provides a simple interface for a highly-scalable managed mobile push notification platform. By using Notification Hubs, an application can send template or personalized notifications across a variety of mobile platforms.

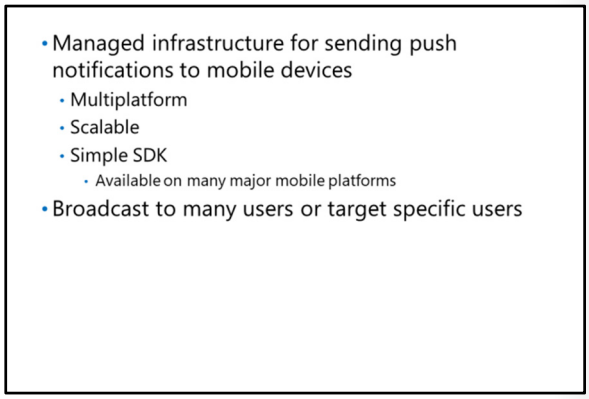
This lesson describes the Notification Hubs service and the methods of integrating with the service.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the architecture of Notification Hubs in Azure.
- Register a client device to a notification hub from either the client or service application.
- Use templates to generalize a notification across platforms.
- Use tags to target notifications to specific client devices.

Service Bus Notification Hubs Overview



Service Bus Notification Hubs is a combination of a service infrastructure and a set of client libraries that allows you to publish push notifications to your mobile applications from any application service component. With Notification Hubs, you can send notifications that are personalized to a specific user, notifications that are distributed to many users across various platforms, and notifications that are filtered to a specific set of users. The Notification Hubs infrastructure abstracts the implementation of the various Platform Notification Systems (PNS) for each mobile platform. By using a single method call, you can send notifications to various device platforms without having to implement a different message structure or communication mechanism for each platform.

You can use notification hubs in a variety of scenarios including:

- Sending wide-reaching news notifications to all devices with your mobile application installed.
- Sending a notification to a subset of your users that is determined based on a tag, label, or location.
- Sending specific notifications to a user for the activities that are related to their specific account.

Notification hubs implement all the functionality of a push infrastructure:

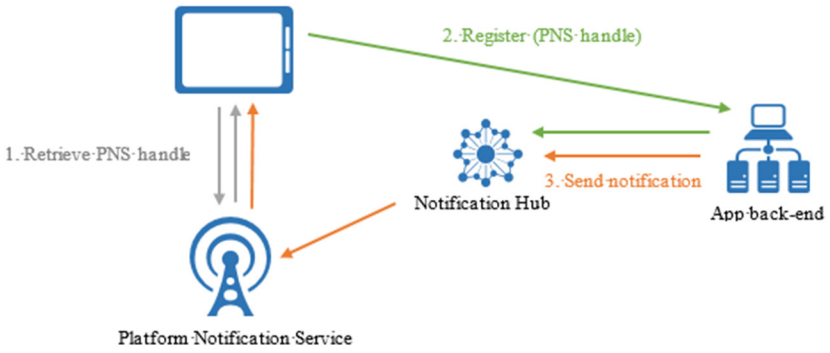


FIGURE 8.4: A NOTIFICATION HUB

Benefits of Using Notification Hubs

Notification hubs eliminate the challenges that are involved in managing push notifications. Notification Hubs use a full multiplatform, scaled-out push notification infrastructure, and considerably reduce the push-specific code that runs in the app. Notification hubs implement all the functionality of a push infrastructure. Devices are only responsible for registering PNS handles, and the backend is responsible for sending platform-independent messages to users or interest groups.

- Managed Infrastructure
  - You don't have to worry about scaling your application yourself
  - You can focus on messages and templates, not the mechanics of your service.
- SDKs available for major platforms
- Template support
- Support for filtering recipients by tag

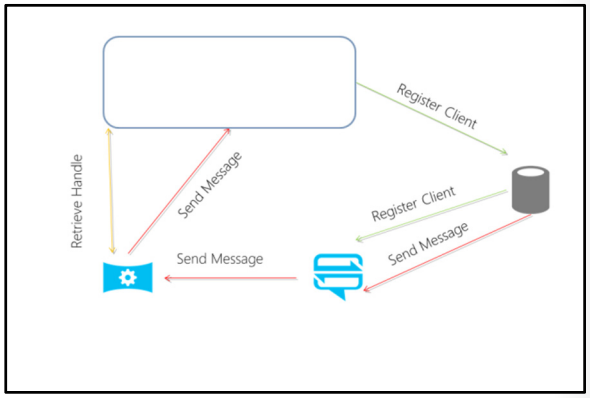
Notification hubs provide a push infrastructure with the following advantages:

- **Multiple platforms**:
  - Support for all major mobile platforms—Windows, Windows Phone, iOS, and Android.
  - No platform-specific protocols. The application only communicates with Notification Hubs.
  - Device handle management. Notification Hubs maintains the handle registry and feedback from PNSs.
- **Works with any backend**. Works with Cloud or on-premises applications that are written in .NET, PHP, Java, or Node.
- **Scale**. Notification hubs scale to millions of devices without the need of rearchitecting or sharding.
- **Rich set of delivery patterns**. Associate devices with tags, representing logical users or interest groups.
  - Broadcast. Allows for near-simultaneous broadcast to millions of devices with a single Application Programming Interface (API) call.

  - Unicast/Multicast. Push to tags representing individual users, including all their devices; or a wider group. For example, a user could use the app on separate devices (tablet, phone, etc.) and would require push notifications to either be pushed to all devices or a specific device.
  - Segmentation. Push to a complex segment that is defined by tag expressions (For example, devices in New York following the Yankees).
- **Personalization**. Each device can have one or more templates to achieve per-device localization and personalization without affecting the backend code.

Notification Hubs Architecture

At a high level, all platform notification systems follow the same pattern:



1. The client application contacts the PNS to retrieve its handle. The handle type depends on the system. For Windows Notification Service (WNS), it is a URI or notification channel. For Apple Push Notification Service (APNS), it is a token.
2. The client application stores this handle in the app backend for later usage. For WNS, the backend is typically a cloud service. For APNS, the system is called a provider.
3. To send a push notification, the app backend contacts the PNS by using the handle to target an instance of a specific client application.
4. The PNS forwards the notification to the device specified by the handle.

Typical PNS infrastructure:

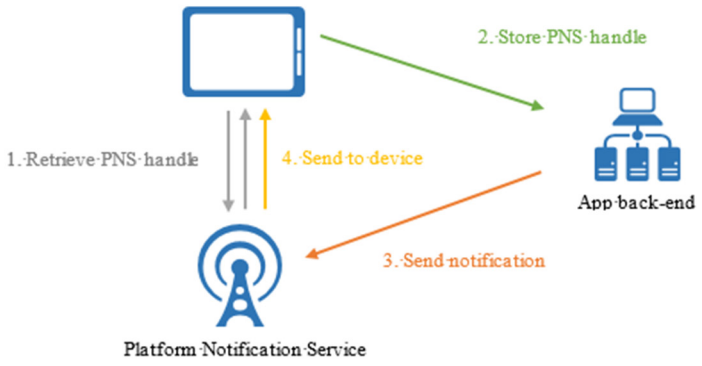


FIGURE 8.5: PNS

With Notification Hubs, you can rely on the service infrastructure to handle the most complex features and have your application focus only on sending messages.

Notification hubs implement all the functionality of a push infrastructure:

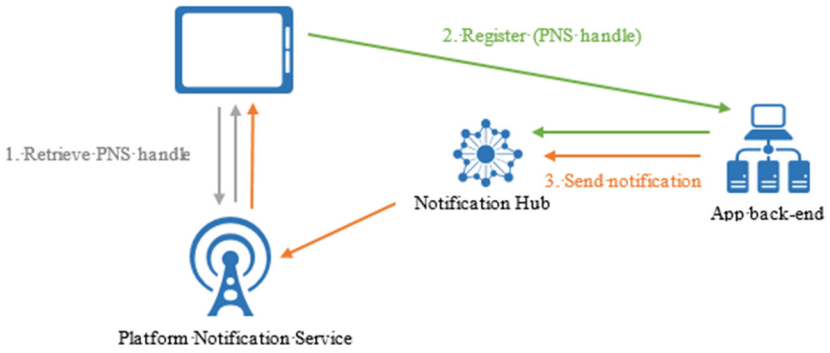


FIGURE 8.6: A NOTIFICATION HUB

Notification Hubs can be used in flexible ways to register devices and eventually send a message to the devices. Devices can register themselves and receive notifications using the following method:

1. The client device reaches out to the PNS by using the Notification Hubs SDK. It registers a unique PNS handle that is used by the service to send notifications to this device whether the application is running or not.
2. The client device can alternatively send its PNS handle to the application backend to have the application register the device.
3. When the application backend sends a message to the Notification Hubs service, the service handles sending the message to the appropriate target clients by using their registered PNS handles. The application backend simply requests the message is sent and the Notification Hubs service and the PNS handle the actual distribution of messages to client devices.

Registrations

A registration is a subentity of a notification hub, and associates a device PNS handle—For example, ChannelURI, device token, or Google Cloud Messaging (GCM) registrationId—with tags and possibly a template. Tags are used to route notifications to the correct set of device handles. Templates are used to implement per-registration transformation.

- Each SDK provides a unique mechanism to register for remote notifications
- You must register with the Notification Hub using the name of the hub and your unique connection string from the connection information panel
  - Two connection strings are available by default:
    - DefaultFullSharedAccessSignature
    - DefaultListenSharedAccessSignature
  - You can opt to use the DefaultListenSharedAccessSignature as a restricted listen-only connection string for your application

Registrations are temporary by default. They can be set to a maximum of 90 days. This can affect the way you design your application and the registration method you select for Notification Hubs. Registrations must contain the most recent PNS handle for each device or channel. Because PNS handles can be obtained only in the client app on the device, one way to manage registrations is to directly register with the Notification Hubs provider on the client app. On the other hand, the security considerations and business logic that is related to the tags might require you to manage the registration in the app backend.

Application Backend Registration

Managing registrations from the backend requires writing additional code. The app from the device must provide the updated PNS handle to the backend every time the app starts (along with tags and templates), and the backend must update this handle on Service Bus. The advantages of managing registrations from the backend are the ability to modify tags to registrations even when the corresponding app on the device is inactive, and to authenticate the client app before adding a tag to its registration. From your app backend, you can perform basic Create, Read, Update and Delete (CRUD) operations on registrations.

Using the application backend to register a device:

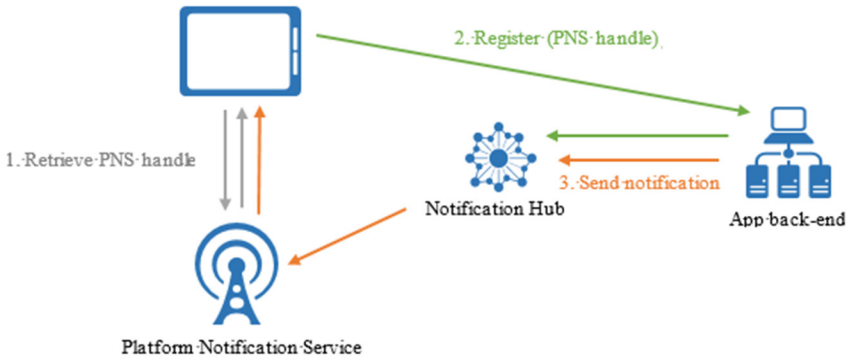


FIGURE 8.7: APPLICATION REGISTRATION

Client Registration

When managing registrations from client apps, the backend is responsible only for sending notifications. Client apps keep the PNS handles up to date and register to tags. The mobile device first retrieves the PNS handle from the PNS, and then registers with the notification hub directly. After the registration is successful, the app backend can send a notification targeting that registration.

The drawback to client registration is that a client app can only update its tags when the app is active. For example, if a user has two devices that register tags that are related to sport teams, when the first device registers for an additional sports team, the second device will not receive the notifications about that team until the app on the second device is run a second time. Generally, when tags are affected by multiple devices, managing tags from the backend is a desirable option.

Registering directly from the client device:

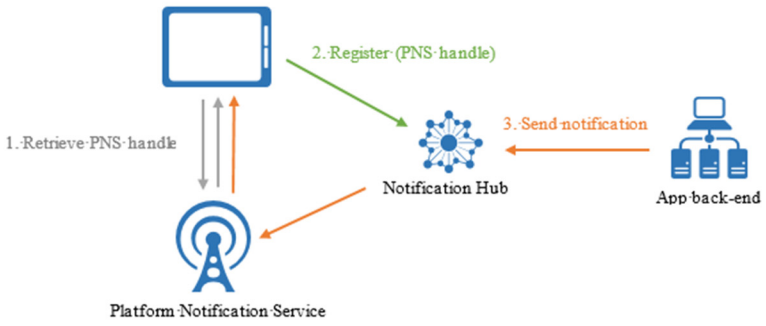


FIGURE 8.8: CLIENT REGISTRATION

Message Templates and Tags

Not all messages sent to a notification hub are intended to be distributed to all device platforms or types and all devices. You can use templates to translate a message into the appropriate structure and format that is compatible with each platform and mobile operating system. You can use tags to filter target devices for a message.

- Templates allow you to send a single message from a back-end and have it transformed into the correctly structured message for each platform
- Templates use a binding format where you can specify where the message will appear in the XML or JSON content
  - Custom properties can be used in the template
- Clients can create multiple registrations to leverage different templates

Templates

Templates enable a client application to specify the exact format of the notifications it wants to receive. For example, the following two payloads are used for Windows and Apple mobile devices.

Notification format for APNS (Apple Platform Notification Service):

APNS JSON payload

```
{“aps”: {“alert” : “Hello!” }}
```

Notification format for WNS (Windows Notification Service):

WNS XML payload

```
<toast>
    <visual>
        <binding template=\"ToastText01\">
            <text id=\"1\">
                Hello!
            </text>
        </binding>
    </visual>
</toast>
```

This requirement forces the app backend to produce different payloads for each platform. This becomes a problem when you consider graphical layouts and localization. The Notification Hubs template feature enables a client app to create special registrations, called template registrations, which include a template in addition to the set of tags. The template is then used to translate a message into the appropriate format or structure for each device.

A template is a set of instructions for the notification hub to format a platform-independent message to the format that is appropriate for each device:

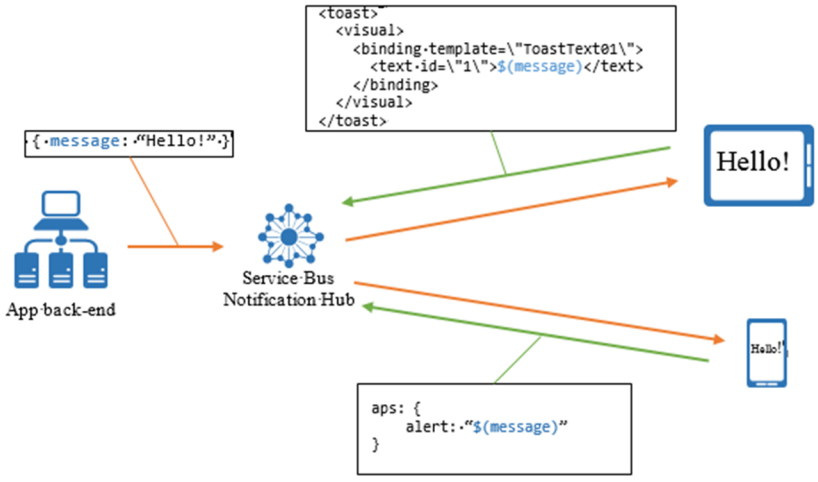


FIGURE 8.9: A TEMPLATE

Tags

Tag expressions enable you to target specific sets of devices, or more specifically registrations, when you send a push notification through Notification Hubs. A tag can be any string, up to 120 characters, that contains alphanumeric characters. The only way to target specific registrations is to associate them with a tag, and then target that tag.

The application backend can choose the registrations to target with a specific notification in the following ways:

- **Broadcast**. All registrations in the notification hub receive the notification.
- **Tag**. All registrations that contain the specified tag receive the notification.
- **Tag expression**. All registrations whose set of tags match the specified expression receive the notification.

Tags do not have to be pre-provisioned and can refer to multiple app-specific concepts. There are cases in which a notification has to target a set of registrations that is identified not by a single tag, but by a Boolean expression on tags. Tag expressions can contain all Boolean operators, such as AND (&&), OR (||), and NOT (!). They can also contain parentheses. Tag expressions are limited to 20 tags if they contain only ORs; otherwise they are limited to six tags.

You can use tag expressions to target many possible matching tags instead of a single tag.

Tag Expression

```
(follows_RedSox || follows_Cardinals) && location_Boston
```

The following example shows an application from which you can receive toast notifications about specific music groups. In this scenario, a simple way to route notifications is to label registrations with tags that represent the different bands:

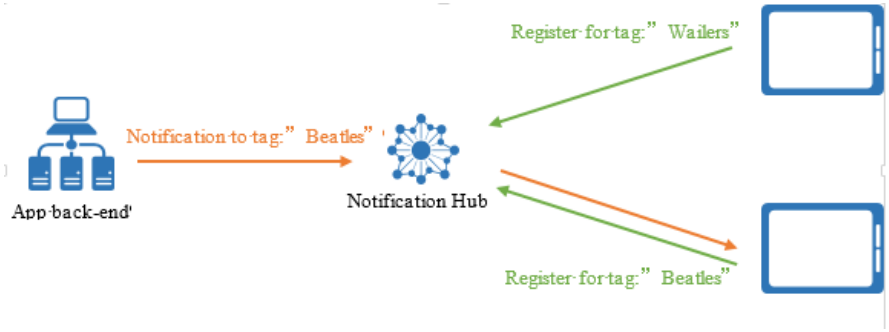


FIGURE 8.10: TAG ROUTING

Lab: Using Queues and Service Bus to Manage Communication in Azure

Scenario

Now that you can generate sign-in sheets in worker roles, you need a scalable and consistent way to enqueue messages for the worker role. You have decided to use an Azure queue mechanism so that you can scale the worker roles in isolation to meet the demand of the queue size. In this lab, you will begin by implementing the communication between the Administration web application and the worker role by using Storage queues. Then you will replace that implementation with an implementation that uses Service Bus queues.

Currently, your on-premises Contoso Events application uses a WCF service to list the hotels that are near a location. You would like to continue to use the WCF service, but you cannot modify your company's firewall. You also would not like to expose the true network location of the WCF service. You have decided to use Service Bus relays so that you have a common endpoint that you can provide to client applications. You will start by using that endpoint in your Contoso Events web application.

Objectives

After you complete this lab, you will be able to:

- Create a Service Bus namespace by using the Management Portal.
- Create Storage queue messages.
- Consume Azure Storage queue messages.
- Create Service Bus queue messages.
- Consume Service Bus queue messages.
- Modify the XML configuration of a WCF service to use the Service Bus relay bindings.
- Modify the C# configuration of a WCF client to use the Service Bus relay bindings.

Lab Setup

Estimated Time: 90 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Creating an Azure Service Bus Namespace

Exercise 2: Using Azure Storage Queues for Document Generation

Exercise 3: Using Service Bus Queues for Document Generation

> **Question:** What are some of the ways that you can automate the change of your WCF bindings from their on-premises versions to a Service Bus Relay binding after the projects are published?

Module Review and Takeaways

In this module, you were introduced to Storage queues. You were also introduced to Service Bus namespaces and the various features that are available for use in your cloud applications. Finally, the two queue mechanisms were compared.

Review Questions

Question: Which queuing mechanism is better suited for storing large messages?

Question: If you have an application that cannot afford to lose any messages, should you use the PeekLock or ReceiveAndDelete mode in your consuming client? How do you check to see if the message is a duplicate?

Question: You have a weather app and you are using Notification Hubs for messaging. You would like to let people within a specific area code receive an emergency alert about a hurricane. How should you architect your application to support this scenario?

Module 9

Automating Integration with Azure Resources

Contents:

Module Overview

Although you can manage most of the Azure services by using both of the Azure portals or Microsoft Visual Studio, you can use scripting to completely automate the management of the same resources. This module will look at automating the lifecycle of the services by using client libraries, Windows PowerShell, REST, and the Resource Manager. Lesson 1, "Creating Azure Scripts by Using Azure PowerShell," describes the modules that are available for managing Azure resources using Azure PowerShell. Lesson 2, "Creating Azure Scripts by Using Azure CLI," describes the cross-platform command-line interface used to manage Azure resources. Lesson 3, "Azure Resource Manager," discusses the Resource Manager architecture in Azure and the concepts associated with this method of managing resources and groups. Lesson 4, "Azure REST Interface," introduces and describes the REST API used to manage all resources in Azure. Lesson 5, "Azure Cloud Shell," describes the Cloud Shell and how it is used to execute scripts within the Azure Portal and context of an Azure subscription.

Objectives

After completing this module, you will be able to:

- Describe the Azure software development kits (SDKs) and client libraries.
- Use Windows PowerShell to automate Azure REST.
- Describe the REST API and the steps to authenticate to the API.
- Use the Resource Manager to create resource groups and templates.

Lesson 1
Creating Azure Scripts by Using Azure PowerShell

You can use Windows PowerShell to automate many of the administration tasks that IT Pros perform on a daily basis. Developers can use Windows PowerShell to facilitate and automate many of their developer operations responsibilities.

This lesson describes the two sets of cmdlets that are available in the Azure PowerShell module.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure PowerShell module.
- List the options for authenticating to Azure by using Windows PowerShell.
- Use Windows PowerShell cmdlets to create an Azure service instance.

The Azure PowerShell Module

The architecture for Microsoft Azure Automation starts with the Microsoft Azure fabric controller. The fabric controller is responsible for finding the resources and then provisioning virtual machines, virtual networks, and almost all of the other backend services in Azure. The REST API is a layer in front of the fabric controller that manages requests from clients and delivers the appropriate requests to the fabric controller. All SDKs including Windows PowerShell and the Cross Platform Command Line Interface are integrated with the REST API. Even the Azure portal uses the same API as the client libraries. Developers can opt to use the REST API directly instead of using a client library or portal.

- Azure PowerShell is a module that is available to manage Azure services
  - Installs a series of cmdLets
- The PowerShell cmdLets provides a superset of the functionality available in the Management Portal.
  - Many times, new features can be implemented using PowerShell long before it is available in the Management Portal.

All client libraries use the REST API:

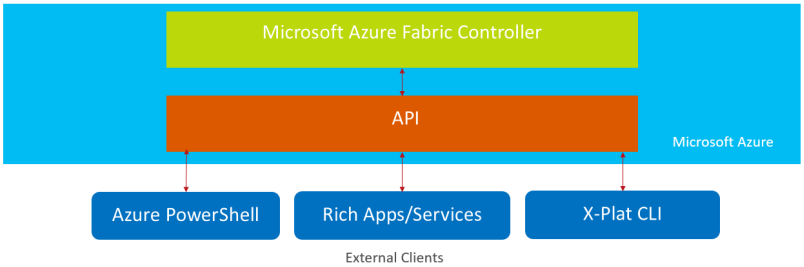


FIGURE 9.1: AZURE AUTOMATION

Azure PowerShell

Azure PowerShell is a collection of two Windows PowerShell modules that you can use to manage Azure services. The first and most commonly used module is the REST module. This module allows you to manage service instances in your Azure subscription. The Resource Manager module is also available and will be discussed in depth in the Azure Resource Manager lesson in this module.

A list of Windows PowerShell activities included in the Azure PowerShell modules is currently available on MSDN:

Reference Link: https://docs.microsoft.com/powershell/azure/overview

One of the major advantages of using the Azure PowerShell modules for REST is the availability of new features. Typically, new features are first introduced to the REST API and then are added to the Azure PowerShell modules. Finally, these features are available in the portal.

Features typically appear in Azure PowerShell prior to the Management Portal:

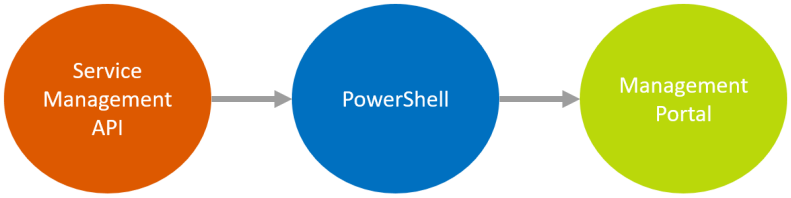


FIGURE 9.2: NEW FEATURES RELEASE ORDER

This typically means that new features can be tested and used by using Azure PowerShell before they are generally available in the portal. Also, you can use Windows PowerShell to create scripts to automate multiple tasks together when you work with the Azure platform. These scripts can be used in developer operations scenarios where configuration needs to be maintained, stored, and repeated.

Azure Automation

Azure Automation is a service that can be used to run Windows PowerShell workflows either on-demand or based on a schedule. The Azure PowerShell activities are already imported into Azure PowerShell and many different Windows PowerShell administration tasks can be directly imported from a local script into Azure Automation. Azure Automation also includes a script center that includes many of the most common administration tasks. These tasks are typically time-consuming, error prone, and complex. With Azure Automation, these scripts can be run as one-off or scheduled jobs and can be run in the Azure environment in an unattended manner.

Reference Link: https://docs.microsoft.com/azure/automation/automation-runbook-gallery

Installing Azure PowerShell Cmdlets

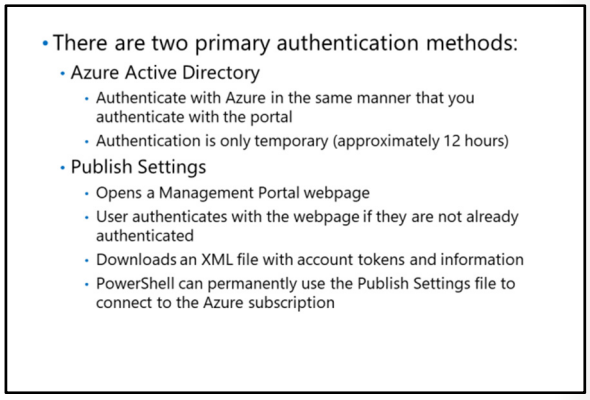
Requirements

Azure PowerShell can be used to manage existing subscriptions but cannot be used to create a new subscription. A subscription is required to use the Azure PowerShell module. Azure PowerShell requires .NET Framework 4.5 to be installed on your local machine. The installer checks for a valid version of Windows PowerShell and then the .NET Framework installs missing dependencies on your machine.

- Azure PowerShell requires .NET 4.5
- A custom PowerShell console is also available that requires less configuration of the environment compared to the standard PowerShell console

Authenticating to Azure by Using Azure PowerShell

To manage the services, the cmdlets need your subscription. There are two ways to provide your subscription information to Windows PowerShell. You can use a management certificate that contains the information, or you can sign in to Azure by using your Microsoft account, a work account, or a school account. When you sign in, Microsoft Azure Active Directory (Azure AD) authenticates the credentials and returns an access token that lets Azure PowerShell manage your account.



PublishSettings File

The **Get-AzurePublishSettingsFile** and **Import-AzurePublishSettingsFile** activities are used to download and import an XML file containing the certificates for your Azure subscriptions. This method of authentication is commonly referred to as certificate authentication. When you use this method, the subscription information is available as long as the subscription and the certificate are valid. However, it is harder to manage access to a shared subscription by using this method. For example, in scenarios when more than one person is authorized to access the account. Also, Azure Resource Manager API does not support certificate authentication.

Azure AD

Azure AD is the recommended authentication method because it makes it easier to manage access to a subscription. The **Add-AzureAccount** activity is used to authenticate in this method. When authenticating by using this method, a dialog box displays and you can sign in by using a Microsoft account or an organizational account in the same manner that you authenticate to the Azure portal.

Demonstration: Managing a Web App by Using PowerShell

Note: To view the latest demo steps, visit the GitHub repository for the course.

Before starting this demo, you must complete the lab in Module 2. For this demo in this module, you will use the available host machine. Also, you must complete the following steps:

1. On the host computer, click **Start**, type **Remote**, and then click **Remote Desktop Connection**.
2. In Remote Desktop Connection, provide the name of your virtual machine in the **Computer** box by using the following format:

   vm20532[*Your Name Here*].cloudapp.net:[*Your VM RDP Port*]

Note: The name and port for your virtual machine might be saved in the Computer drop-down list. If this is the case, use this value instead of typing it in manually. If you are unsure about your virtual machine's RDP port, use either of the Azure portals to find your virtual machine's endpoints. The endpoint with the name **Remote Desktop** is the correct port for RDP. This port is randomized to protect your virtual machine from unauthorized access.

3. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.

4. If necessary, sign in by using the following credentials:
    - User name: **Student**
    - Password: **AzurePa$$w0rd**

Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Lesson 2
Creating Azure Scripts by Using Azure CLI

The Azure CLI 2.0 is Azure's new command-line experience for managing Azure resources. You can use it in your browser with Azure Cloud Shell, or you can install it on macOS, Linux, and Windows and run it from the command line.

This lesson introduces the Azure CLI.

Lesson Objectives

After completing this lesson, you will be able to:

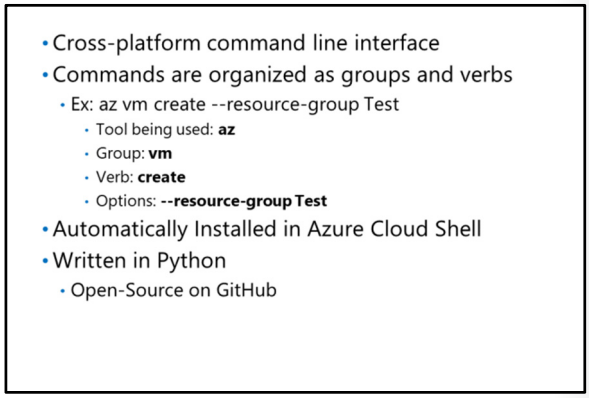
- Describe the Azure CLI and its command structure.
- Install the Azure CLI.
- Use the Azure CLI in interactive mode.

Azure CLI 2.0

Azure CLI 2.0 is optimized for managing and administering Azure resources from the command line, and for building automation scripts that work against the Azure Resource Manager.

Commands in the CLI are provided as subcommands of groups. Each group represents a service provided by Azure, and the subgroups divide commands for these services into logical groupings.

Example Azure CLI command:



az vm create

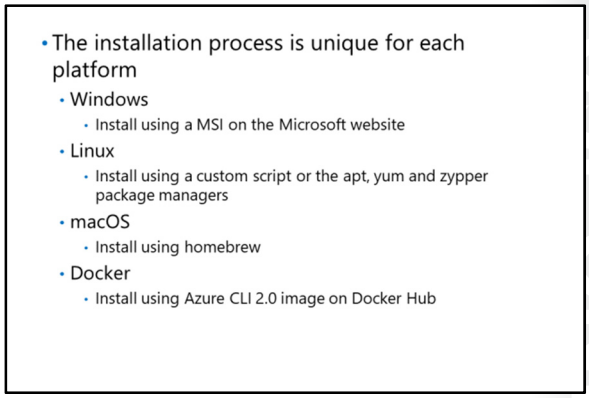
```
az webapp create -g MyResourceGroup -p MyPlan -n MyUniqueAppName
```

To search for commands, use az find. For example, to search for command names containing secret, use the following command:

az find

```
az find -q secret
```

Installing the Azure CLI



Windows

On Windows the Azure CLI binary is installed via an MSI, which gives you access to the CLI through the Windows Command Prompt (CMD) or PowerShell. If you are running Windows Subsystem for Linux (WSL), there are packages available for your Linux distribution.

Linux

Before installing the Azure CLI on your Linux distribution, you should first check to see which package manager is used by your distribution. Azure CLI packages are available for the **apt**, **yum** and **zypper** package managers. If you do not have a package for the Azure CLI available on your distribution, you can always install the CLI manually by running an installation script.

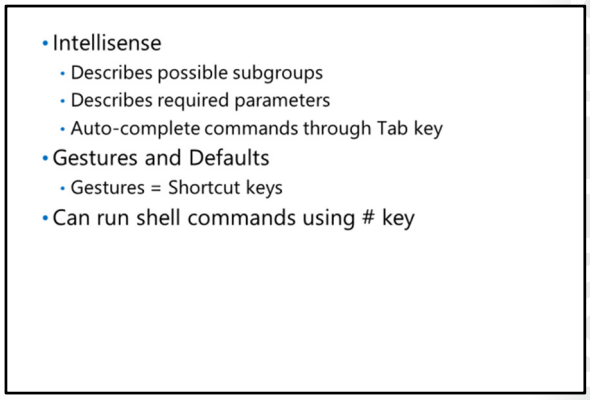
macOS

For the macOS platform, you can install the Azure CLI with homebrew package manager. Homebrew is the easiest way to manage your CLI install. It provides convenient ways to install, update, and uninstall.

Docker

You can use Docker to run a standalone Linux container with the Azure CLI 2.0 pre-installed. Docker lets you get started quickly with an environment where you can try out the CLI to decide if it's right for you, or use our image as a base for your own deployment.

CLI Interactive Mode



You can use Azure CLI 2.0 in interactive mode by running the az interactive command. That places you in an interactive shell where your commands are auto-completed and you have access to descriptions of commands and their parameters and command examples.

Interactive mode optionally displays command descriptions, parameter descriptions, and command examples. You can turn descriptions and examples on or off using the F1 key.

You can also run shell commands without leaving interactive mode using **#[cmd]**.

Lesson 3
Azure Resource Manager

With the new Azure Preview Portal (Ibiza), a new way of managing resources in Azure has emerged. With resource groups and resource group templates, automating the creation and monitoring of multiple-service workloads is made much easier.

This lesson describes the new Resource Manager offering.

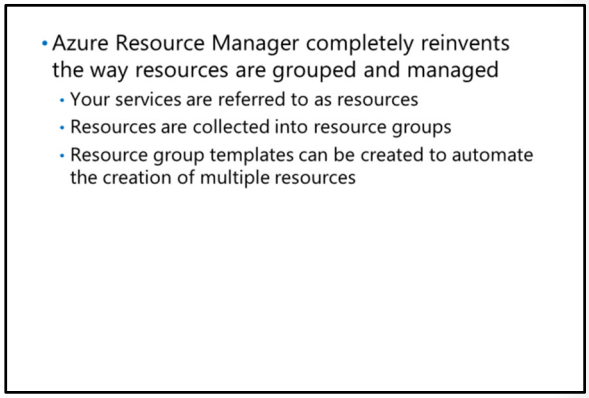
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure Resource Manager.
- Describe resource groups and resource group templates.
- Use the Resource Manager to create multiple resources as a single managed unit.

Azure Resource Manager Overview

Resource Manager introduces an entirely new approach to Azure resources. Service instances are now referred to as Resources, which can be logically stored into resource groups. Resource groups provide a common lifecycle for the child resources. They can be created, managed, monitored, or deleted together. The Resource Manager also offers the concept of resource group templates which enable you define a service unit in advance, and then use the template to create as many resource groups as you need.



Resource groups and resource group templates are ideal for developer operations scenarios where you need to quickly build out development, test, quality assurance, or production environments that are homogenous in nature and can be managed with a shared lifecycle. Developers can quickly delete their environment and create a new environment by using the shared template. The resource groups can be monitored to determine the billing rate or resource usage at a higher level than monitoring individual service instances.

The Resource Manager functionality in Azure is new and is visible only in the Preview Portal. A set of Windows PowerShell cmdlets are available to manage resource groups today. The functionality is limited however and only a subset of the Azure services can be managed in resource groups by using Windows PowerShell and the Azure portals. The following are some of the current limitations:

- All services are not available in resource groups. For example, API Management instances cannot be added to a resource group.
- Resource group templates can only be used to manage some services and cannot be used with Virtual Machines yet.

However, new Resource Manager functionality is being released on a weekly basis and these limitations are likely to change quickly.

Resource Groups

- Resource groups are collections of related resources
  - By managing the resources as a group, you can create, modify, or destruct the resources as a whole
  - Resources can be migrated and managed as a logical unit
- Resource groups are often referred to as lifecycle boundaries since they now can share a managed service lifecycle
- Resource groups can also be used to track usage and billing isolated to a specific logical unit

Whenever a resource is created in the Preview Portal, it is always created within a resource group. You can choose to create a new resource group or use an existing resource group when creating the service instance. You might also notice that some resources that are created in the Management Portal are also placed into resource groups. These are only visible in the Preview Portal.

When you create an application that consists of a few resources working together it is always created in its own resource group, so that you can manage the lifecycle of all related assets by using the resource group. You can add or remove additional resources from the resource group as your application evolves. For example, the **Website + SQL** option in the Preview Portal creates a new resource group that consists of a website, a web hosting plan, and a SQL database and server along with other resources.

Resource groups can be viewed either by using the Preview Portal or Azure PowerShell. By using the Preview Portal, resource groups can be viewed and monitored as a group.

A resource group has a blade that provides you with all the information about a particular resource group:

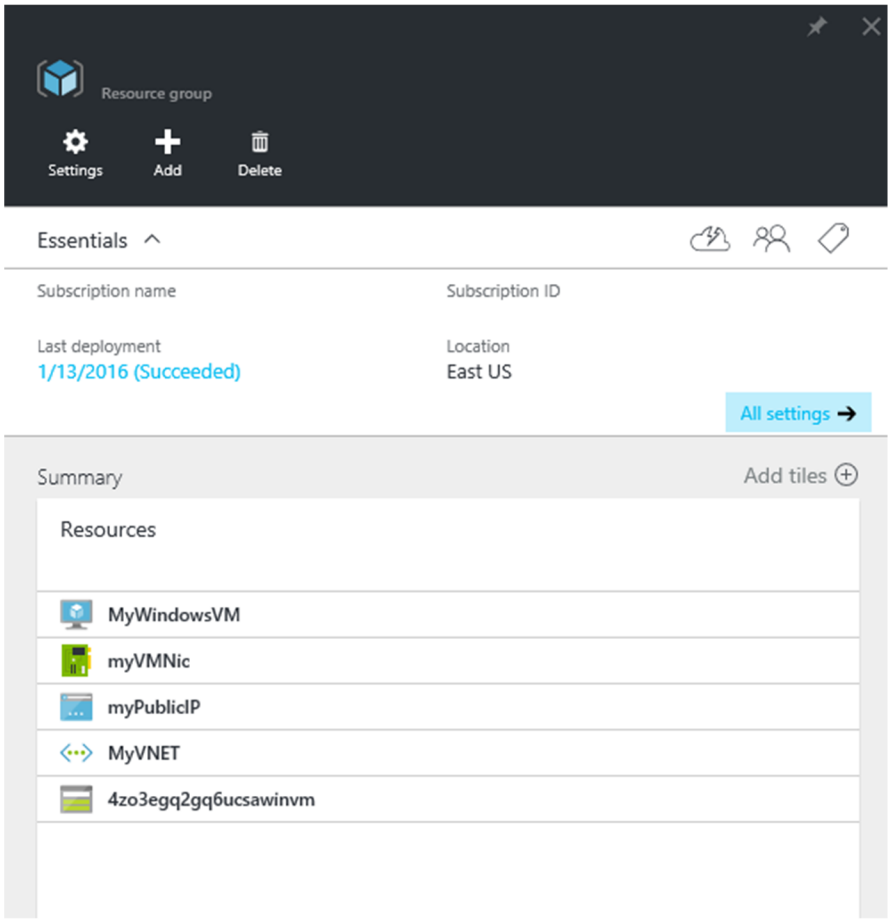
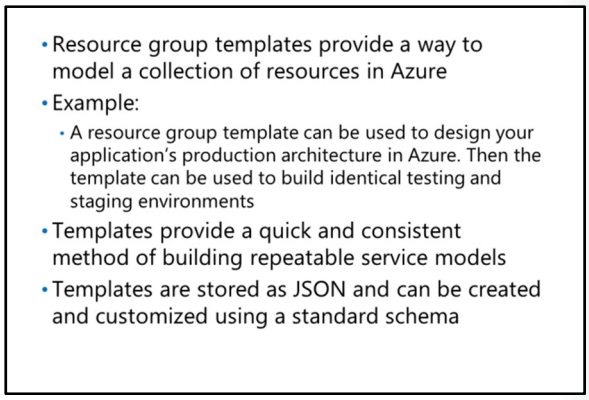


FIGURE 9.3: RESOURCE GROUP BLADE

Resources can be added to a resource group at any time. The Preview Portal has an Add button that can be used to add a new resource to a resource group. Resource groups also enable you to manage the lifecycle of all the contained resources. Deleting a resource group will delete all the resources contained within it.

The Resource Manager mode in Azure PowerShell will allow you to manage resource groups in your Azure subscription. Resource groups are created by using the **New-AzureResourceGroup** cmdlet. You can create a resource group with a name and a location, and then use the **New-AzureResource** cmdlet to manually create resources and add them to the resource group. You can also use a resource group template to create a resource group from an existing definition.

Resource Group Templates



Most applications that are designed to run in Azure use a combination of resources, such as a database server, a database, and a website, to perform as designed. An Azure Resource Manager template makes it possible for you to deploy and manage these resources together by using a JavaScript Object Notation (JSON) description of the resources and associated deployment parameters. There are several ways to create a resource group and its resources, but the easiest way is to use a resource group template. A resource group template is a JSON string that defines the resources in a resource group. The string includes placeholders called parameters for user-defined values, such as names and sizes.

The template consists of a variety of sections including those listed in the following table.

$schema	A schema file must be specified indicating the version of the template language that should be used.
parameters	Parameters can be specified in a template so that the same template can be used for multiple resource groups. For example, a resource group template can create a hosting plan and website. By using parameters, the same template can be used to create either a Standard tier hosting plan or a Free tier hosting plan.
variables	Variables are reusable data that can be used for resources in the template. This can cut down on the amount of repeated content in the template and honors the DRY (don't repeat yourself) principles of software development.
resources	The resources section is a JSON array of the individual resources that are defined in your template. This section is hierarchical and can be defined in a manner such that certain resources are created first because of dependencies. For example, a website resource in a template can have a web hosting plan resource nested in its definition. This will ensure that the web hosting plan is created before the website instance is created.

The resource group template JSON schema (language) also allows you to specify output values when the resource group is created and the functions that can be used throughout your template.

Azure hosts a gallery of resource group templates and you can create your own templates, either from scratch or by editing a gallery template. Existing resource group templates can be downloaded by using Azure PowerShell. The **Save-AzureResourceGroupGalleryTemplate** cmdlet downloads a template from the gallery to a JSON file.

The **Save-AzureResourceGroupGalleryTemplate** cmdlet allows you to view existing gallery resource group templates.

Save-AzureResourceGroupGalleryTemplate

```
Save-AzureResourceGroupGalleryTemplate -Identity Microsoft.WebSiteSQLDatabase.0.2.2-
preview -Path D:\Azure\Templates
```

Example resource group template for a website:

Resource Group Template JSON

```
{
    "parameters": {
        "siteName": {
            "type": "string"
        },
        "hostingPlanName": {
            "type": "string"
        },
        "siteLocation": {
            "type": "string"
        },
        "sku": {
            "type": "string",
            "allowedValues": [
                "Free",
                "Shared",
                "Basic",
                "Standard"
            ],
            "defaultValue": "Free"
        }
    },
    "resources": [
        {
            "name": "[parameters('siteName')]",
            "type": "Microsoft.Web/Sites",
            "location": "[parameters('siteLocation')]",
            "dependsOn": [
                "[concat('Microsoft.Web/serverFarms/', parameters('hostingPlanName'))]"
            ],
            "properties": {
                "sku": "[parameters('sku')]",
                "name": "[parameters('siteName')]"
            }
        }
    ]
}
```

Demonstration: Viewing a Resource Group Template

Note: To view the latest demo steps, visit the GitHub repository for the course.

Before starting this demo, you must complete the lab in Module 2. For this demo in this module, you will use the available host machine. Also, you must complete the following steps:

1. On the host computer, click **Start**, type **Remote**, and then click **Remote Desktop Connection**.
2. In Remote Desktop Connection, provide the name of your virtual machine in the **Computer** box by using the following format:

   vm20532[*Your Name Here*].cloudapp.net:[*Your VM RDP Port*]

Note: The name and port for your virtual machine might be saved in the Computer drop-down list. If this is the case, use this value instead of typing it in manually. If you are unsure about your virtual machine's RDP port, use either of the Azure portals to find your virtual machine's endpoints. The endpoint with the name **Remote Desktop** is the correct port for RDP. This port is randomized to protect your virtual machine from unauthorized access.

3. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.
4. If necessary, sign in by using the following credentials:
   - User name: **Student**
   - Password: **AzurePa$$w0rd**

Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Lesson 4
Azure REST Interface

Many modern services and applications provide REST APIs. HTTP is a universal standard and can be used across a variety of platforms and languages. Azure provides a REST API that can be used to manage services regardless of your current platform.

This lesson describes the Azure REST API and how to authenticate against the API.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the REST API.
- List the two options that are used for authenticating to the API.

Azure REST API

The Azure API provides programmatic access to most of the functionality that is available through Azure PowerShell or the Management Portal. The API is a REST API with operations for common HTTP methods such as **GET**, **PUT**, **POST**, or **DELETE**. All API operations are performed over Secure Sockets Layer (SSL) by using X.509 certificates.

- Azure Resources can be managed using a REST API and common HTTP verbs
  - API operations are performed using SSL
  - REST APIs are supported by a variety of programming and scripting platforms
- Many of the same features from the portal can be used with the REST API

Each subscription in Azure is assigned a unique subscription ID. All resources and resource groups in Azure are associated with a subscription and are referenced by using the subscription ID. The subscription ID also is part of the URI for every call that is made by using the REST API.

Reference Link: https://docs.microsoft.com/rest/api/

Authenticating Requests to the Azure REST API

There are two primary methods of authenticating requests to the Azure REST API. Azure Active Directory can be used for authentication from custom applications. Management certificates can be used for authentication for management tasks from a variety of other clients.

- Authentication to the REST API can be done in two ways:
  - Using Azure Active Directory
    - Add your calling application as an application in Azure AD. You can then delegate permissions to your application to use the REST API.
  - Using Management Certificate
    - An X509 certificate can be downloaded and used as part of an HTTP request to access the REST API

Management Certificate

Secure requests to the management service can be authenticated by using management certificates over SSL. To use a management certificate, it must be uploaded to Azure. After

you add a management certificate to the subscription, you can sign the requests to the service by using the same certificate. The REST API does not verify if a certificate is still valid and therefore it can be used with an expired or invalid certificate. Also, role-based authentication is not supported for certificate-based authentication.

Azure Active Directory

Secure requests to the management service can be authenticated by creating an Azure AD application and using the Active Directory Authentication Library to obtain an access token from the application.

Reference Link: https://docs.microsoft.com/dotnet/api/microsoft.identitymodel.clients.activedirectory?view=azure-dotnet

The Azure AD Authentication Library for .NET enables client application developers to easily authenticate users to a cloud or on-premises Active Directory, and then obtain access tokens for securing API calls. Active Directory Authentication Library for .NET has many features that make authentication easier for developers. Some examples are asynchronous support, a configurable token cache that stores access and refresh tokens, and automatic token refresh when an access token expires and a refresh token is available. By managing these complex tasks, your application can focus on the relevant code and not on authenticating with the API.

After you create a native client application in Azure AD, you need to assign delegated permissions to access the REST API, and then use the **clientId** and **tenantId** parameters in your custom application.

Azure SDKs

Language-specific tools and client libraries are available for a variety of platforms. You can use these tools and libraries when you integrate your custom applications with Azure.

- Software Development Kits (SDK) are available for different platforms
  - It provides programmatic access to Azure resources
- Source code for SDKs are open-source and available on GitHub
- Official Azure tools are available for Eclipse and Visual Studio

Microsoft Platform

To manage Azure services, extensions are available for Visual Studio 2012 and later that enhance the functionality of the Server Explorer and adds new project templates. Visual Studio allows you to manage Azure services and display their status using the Server Explorer. In Visual Studio, you can also create new services and manage current services either using the Server Explorer or the project templates. For more complex integration with your custom applications, .NET libraries are available to manage Azure services. Many of these custom packages are available directly on NuGet.

Reference Link: https://www.nuget.org/packages?q=windowsazureofficial

For the most complex administration scenarios, Windows PowerShell activities are available to automate your activities with Windows PowerShell scripts. You can install these Windows PowerShell activities directly in your development environment by using the Azure PowerShell modules. Alternatively, the Azure PowerShell modules are available in the Azure Automation service.

Reference Link: https://docs.microsoft.com/dotnet/azure/

Third-Party Platforms

Similar Azure REST libraries are available for popular platforms and languages including:

- Java
- Node.js
- PHP
- Python
- Ruby

This list is expanding on a fast cadence. For the majority of these languages or frameworks, installers are available for multiple popular operating systems (Windows, Mac OS, and Linux). Developer portals are also available on the official Azure website with links to tutorials and advanced documentation.

A set of open-source commands are available in a collection referred to as the Azure Cross-Platform Command-Line Interface (xplat-cli). Xplat-cli provides a common interface for managing Azure services regardless of the operating system or management environment. Xplat-cli is written in Node.js and requires a local installation of Node.

Reference Link: https://azure.microsoft.com/develop/php/

Reference Link: http://azure.microsoft.com/develop/nodejs/

Reference Link: http://azure.microsoft.com/develop/java/

Reference Link: http://azure.microsoft.com/develop/python/

Reference Link: http://azure.microsoft.com/develop/ruby/

Mobile Platforms

Microsoft Azure Mobile Services is a platform that enables you to quickly build a back-end service for mobile applications. With a dynamic schema, this mobile back-end service can be created with little or no custom code and can be refined over time. Mobile Services and its dynamic schema feature allows developers to bring mobile applications and ideas to market quickly without many of the traditional delays and infrastructure setup. Mobile Services also scales with the application and can handle growth in usage for the application over time.

Mobile Services can integrate with a variety of mobile platforms with native libraries including:

- iOS
- Android
- Windows Phone 8
- Windows Store (C# or JavaScript)

Mobile Services also exposes a RESTful HTTP endpoint and supports cross-origin resource sharing (CORS). This ensures that Mobile Services can be used for a wide variety of scenarios where a native library is not immediately available.

Reference Link: https://docs.microsoft.com/azure/app-service-mobile/app-service-mobile-value-prop

Media Services

Microsoft Azure Media Services is an all-encompassing media solution that includes the encoding, storage, protection, and delivery of media assets to client devices. Libraries are available for multiple media platforms such as Silverlight or Flash, development platforms such as .NET or Java, and platforms such as Windows 8, Windows Phone, Android and iOS.

Reference Link: https://docs.microsoft.com/azure/media-services/media-services-overview

Lesson 5
Azure Cloud Shell

Azure Cloud Shell is an interactive, browser-accessible shell for managing Azure resources. It provides the flexibility of choosing the shell experience that best suits the way you work. Linux users can opt for a Bash experience, while Windows users can opt for PowerShell.

This lesson briefly introduces the Cloud Shell.

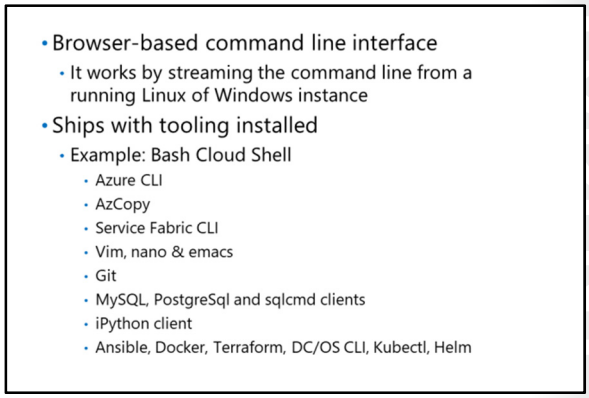
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Azure Cloud Shell.
- Configure the Cloud Shell to store files between runs.

Cloud Shell

Cloud Shell enables access to a browser-based command-line experience built with Azure management tasks in mind. Leverage Cloud Shell to work untethered from a local machine in a way only the cloud can provide.



Azure Cloud Shell is assigned per unique user account and automatically authenticated with each session. Get a modern command-line experience from multiple access points, including the Azure portal, shell.azure.com, Azure mobile app, Azure documentation, and VS Code Azure Account extension.

Azure Cloud Shell gives you the flexibility of choosing the shell experience that best suits the way you work. Linux users can opt for a Bash experience, while Windows users can opt for PowerShell. Microsoft routinely maintains and updates Cloud Shell, which comes equipped with commonly used CLI tools including Linux shell interpreters, PowerShell modules, Azure tools, text editors, source control, build tools, container tools, database tools, and more. Cloud Shell also includes language support for several popular programming languages such as Node.js, .NET, and Python.

Cloud Shell attaches an Azure Files share to persist your data. On first use, Cloud Shell will prompt to create a file share in Azure Files (or attach an existing one) to persist your data across sessions, and Cloud Shell will automatically re-attach it for subsequent sessions.

Lab: Automating the Creation of Azure Assets by Using PowerShell and Azure CLI

Scenario

Now that you have created many of the resources that you will use in your Azure application, you have decided to automate the creation of your assets in Azure. Some of your administrators are Windows experts and would prefer to automate using PowerShell while others use Linux and would prefer to automate from the command line. Due to this requirement, you will try and implement automation using PowerShell and separately using a cross-platform CLI interface.

Objectives

After you complete this lab, you will be able to:

- Authenticate to Azure by using xPlat CLI.
- Create a Website instance by using xPlat CLI.
- Remove a Website instance by using xPlat CLI.
- Authenticate to Azure by using PowerShell.
- Create a resource group from a template by using PowerShell.
- Remove a resource group by using PowerShell.

Lab Setup

Estimated Time: 45 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Use Azure CLI to Create and Manage an Azure Web App

Exercise 2: Use PowerShell to Create and Manage an Azure Storage Account

Question: When automating your Azure accounts and resource management, should you use the Resource Manager cmdlets or the standard cmdlets?

Check Your Knowledge

Question	
Which Azure cmdlet can be used to verify that a resource group has been deleted?	
Select the correct answer.	
	Remove-AzureRmResourceGroup
	Get-AzureRmResourceStatus
	Get-AzureRmResourceGroup
	Delete-AzureRmResourceGroup
	Get-AzureRmResourceGroupTemplate

Module Review and Takeaways

In this module, you were introduced to the various methods of automating the management aspects of Azure. Although SDKs, client libraries, xplat CLI, and the REST APIs were discussed, significant focus was placed on automation by using Windows PowerShell. Windows PowerShell can be used to manage services by using both the REST cmdlets or the Azure Resource Manager cmdlets.

Best Practice

Moving forward, the Azure Resource Manager is the preferred method of managing resources in Azure. Resource should be grouped into logical units by using resource groups and templates should be created for resources that might need to be created multiple times.

Review Question

Question: How can you create your own resource group template?

Module 10

DevOps in Azure

Contents:

Module Overview

Although you can deploy your cloud applications manually, it is in your best interest to begin automating cloud-based deployments. Automation creates many benefits including the ability to trace past actions, easier repetition of deployment tasks and reduced possibility of human error. Lesson 1, "Continuous Integration," discusses strategies for integrating source control repositories with running cloud service instances for automatic deployment scenarios. Lesson 2, "Azure DevTest Labs," introduces the DevTest service which is useful for automating the creation of machine-specific environments and lab scenarios. Lesson 3, "Azure Resource Manager Templates," discusses the capability to deploy entire workloads in Azure from a JSON template. Lesson 4, "Managed Solution Hosting," introduces Service Fabric, Azure Container Service and Azure Container Instances as methods used to host solutions using a fully managed service.

Objectives

After completing this module, students will be able to:

- Automate the deployment of their applications to Azure.

Lesson 1
Continuous Integration

The process of Continuous Integration (CI) encourages developers to frequently integrate their code. It also provides the benefits of using automated build and testing processes.

This lesson introduces Continuous Integration conceptually as a context for later lessons in this module.

Lesson Objectives

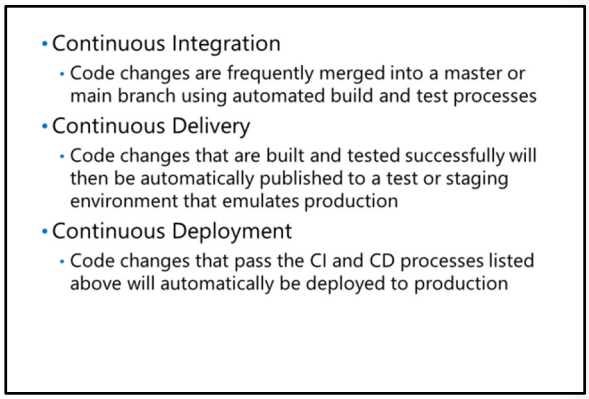
After completing this lesson, you will be able to:

- Describe the differences between CI and CD.

Continuous Deployment

Continuous integration and continuous delivery (CI/CD) are a key requirement for achieving success with the cloud. Without a good CI/CD process, you will not achieve the agility that the cloud promises.

When we talk about CI/CD, we are really talking about several related processes: Continuous integration, continuous delivery, and continuous deployment.



- Continuous integration means that code changes are frequently merged into the main branch, using automated build and test processes to ensure that code in the main branch is always production-quality.
- Continuous delivery means that code changes that pass the CI process are automatically published to a production-like environment. Deployment into the live production environment may require manual approval, but is otherwise automated. The goal is that your code should always be *ready* to deploy into production.
- Continuous deployment means that code changes that pass the CI/CD process are automatically deployed into production.

Lesson 2
Azure DevTest Labs

Developers and testers are looking to solve the delays in creating and managing their environments by going to the cloud. Azure solves the problem of environment delays and allows self-service within a new cost-efficient structure. However, developers and testers still need to spend considerable time configuring their self-served environments.

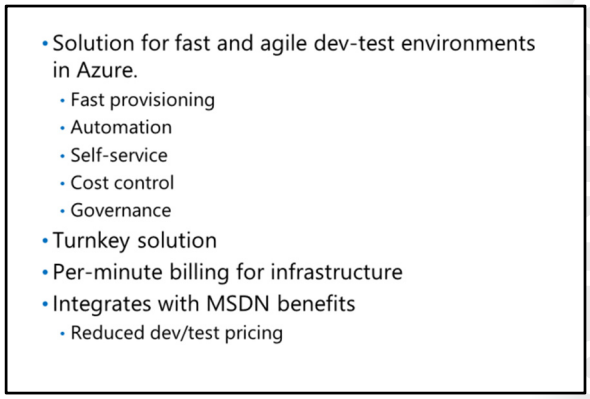
This lesson proposes the Azure DevTest Labs service as a solution to managing the creation of multiple environments for the validation of applications.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the DevTest Labs service.
- Use VSTS to create a CI/CD pipeline using DevTest Labs.

DevTest Labs



In today's world developers have the need to setup their solution environments in the quickest way possible and without any delay. Microsoft Azure is a great solution for this use case. Developers can create the needed environments with all the big advantages that a Cloud solution provides. Nevertheless, this is a task that need special skills that in some companies are reserve for IT Pros and when they are done by developers these tasks usually take some time. Time that the development team is moving from the current project to the creation of the environment, so, instead of being focus on the project they are creating the solution environments. The Microsoft Azure DevTest Labs comes to rescue helping developers to quick create the needed environments in a very short period, minimizing the time the development team is not focus of the current project. Azure DevTest Labs can create Windows and Linux Virtual Machines based in reusable templates. With the use of their templates, developers can create and remove environments wherever they need to test a new feature, a new solution, etc. These deployments can also be integrated in the DevOps pipeline in order to create environments for the Test teams, for automatically tests, etc.

Azure DevTest can also be used to create Training and demonstration environments of the solution being developed. This also solves issues of the teams that need to create these environments due to the flexibility of the service.

Key Concepts

Lab. Group of resource that consist or lab can be several virtual machines.

Virtual Machine. Azure Virtual Machine based on an image.

Claimable VM. A claimable VM is unassigned and can be claimed by any lab user with permissions. The lab administrator can mark the virtual machines as claimable in the advanced settings blade, then they will show up under the claimable virtual machines list in the overview blade.

Base Image. You can create a custom base image, creating in this way an already pre-configured virtual machine with all the software and configuration needed for a specific deployment. This way is faster to setup a lab.

Artifacts. Based on JSON files providing common tools to be installed in an Azure DevTest Virtual Machines. You can find tools like, Atom, Google Chrome, 7-Zip, git, Node JS, Notepad++, … You can also use artifacts to setup your firewall rules, create an Active Directory domain or even to clone a specific repo.

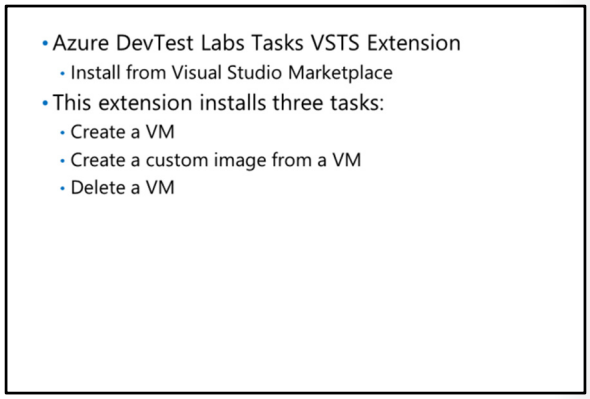
Formulas. DevTest Lab template to fully setup a virtual machine with a based image and all the needed software installed via selection of the appropriate artifacts.

You start to select a base image of your virtual machine, then you can setup the all Virtual machine configurations like: name, User Name, virtual machine size and all the artifacts that this formula will have. Using formulas, you can easily create a developer environment with all the software that your developers need on the machine without the need to have then install and configure the machine.

Policies. In Azure DevTest Labs policies can be used to control the cost of a lab. You can create a specific Policy specifying limits and quotas. For example, you can configure a schedule to shut down your virtual machines, limit the number of virtual machines per user and per lab or set an expiration date on your virtual machine.

Visual Studio Team Services CI/CD Integration

Azure DevTest Labs is the perfect Azure service to integrated with Visual Studio Team Services (VSTS). We can use DevTest Labs to create application development and test environments, but with the use of the Azure DevTest Labs Extension you can integrate the creation of the environment (Lab) in the release pipeline. You can install this extension in your VSTS Tenant at the following link.



Reference Link: https://marketplace.visualstudio.com/items?itemName=ms-azuredevtestlabs.tasks

Lesson 3
Azure Resource Manager Templates

With Resource Manager, you can create a template (in JSON format) that defines the infrastructure and configuration of your Azure solution. By using a template, you can repeatedly deploy your solution throughout its lifecycle and have confidence your resources are deployed in a consistent state.

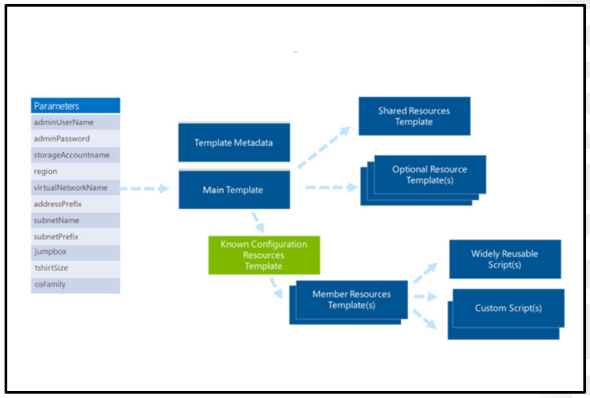
This lesson introduces ARM templates and describes how to author and deploy the templates.

Lesson Objectives

After completing this lesson, you will be able to:

- Author an Azure Resource Manager template.
- Deploy an ARM template.
- Use an ARM template as a solution in the Azure Marketplace.

Resource Manager Templates



Most applications that are designed to run in Azure use a combination of resources, such as a database server, a database, and a website, to perform as designed. An Azure Resource Manager template makes it possible for you to deploy and manage these resources together by using a JavaScript Object Notation (JSON) description of the resources and associated deployment parameters. There are several ways to create a resource group and its resources, but the easiest way is to use a resource group template. A resource group template is a JSON string that defines the resources in a resource group. The string includes placeholders called parameters for user-defined values, such as names and sizes.

The template consists of a variety of sections including the following.

$schema	A schema file must be specified indicating the version of the template language that should be used.
parameters	Parameters can be specified in a template so that the same template can be used for multiple resource groups. For example, a resource group template can create a hosting plan and website. By using parameters, the same template can be used to create either a Standard tier hosting plan or a Free tier hosting plan.
variables	Variables are reusable data that can be used for resources in the template. This can cut down on the amount of repeated content in the template and honors the DRY (don't repeat yourself) principles of software development.
resources	The resources section is a JSON array of the individual resources that are defined in your template. This section is hierarchical and can be defined in a manner such that certain resources are created first because of dependencies. For example, a website resource in a template can have a web hosting plan resource nested in its definition. This will ensure that the web hosting plan is created before the website instance is created.

The resource group template JSON schema (language) also allows you to specify output values when the resource group is created and the functions that can be used throughout your template.

Template Advantages

Templates are generally preferred to manually deploying resources for quite a few reasons:

- A template can ensure idempotency. If you deploy an identical template to multiple resource groups, they would functionally be the same.
- A template can simplify orchestration as you only need to deploy the template to deploy all of your resources. Normally this would take multiple operations.
- A template allows you to configure multiple resources simultaneously and use variables/parameters /functions to create dependencies between resources. For example, you can require that a VM is created before a Web App because you need the VM's public IP address for one of the Web App's settings. Another example is to require a Storage account is created before a VM so that you can place the VHDs in that storage account.
- A template is a JSON file so it can be compared, managed using a source control provider and used as part of any continuous integration process.
- Templates can parameterize input and output values so they can be reused across many different scenarios. Templates can also be nested so you can reuse smaller templates as part of a larger orchestration.

Templates make it easier to repeat and reuse configurations throughout your solutions:

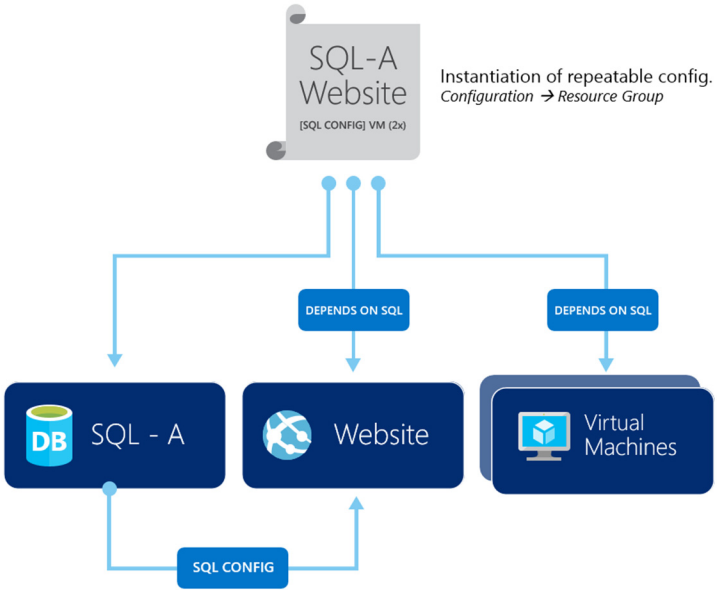


FIGURE 10.1: ARM TEMPLATE EXAMPLE

Deploying ARM Templates

When deploying resources to Azure, you:

1. Log in to your Azure account.
2. Create a resource group that serves as the container for the deployed resources. The name of the resource group can only include alphanumeric characters, periods, underscores, hyphens, and parenthesis. It can be up to 90 characters. It cannot end in a period.
3. Deploy to the resource group the template that defines the resources to create.

- Using the Portal
  - https://portal.azure.com/#create/Microsoft.Template
- From a Hyperlink
  - https://portal.azure.com/#create/Microsoft.Template/uri/[URL encoded URI to ARM Template]

A template can include parameters that enable you to customize the deployment. For example, you can provide values that are tailored for a particular environment (such as dev, test, and production). The Resource Manager template you deploy can either be a local file on your machine, or an external file that is located in a repository like GitHub.

Lesson 4
Managed Solution Hosting

Many application solutions require some type of hosting that has the advantage of minimal maintenance (similar to App Services) but also has the flexibility and power to host complex solutions that have requirements that don't "cleanly" fit into the App Services sandbox.

This lesson will show how you can host applications on Service Fabric, Container Service and Container Instances.

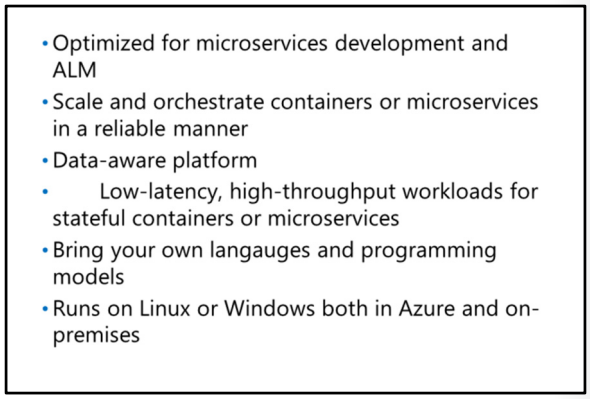
Lesson Objectives

After completing this lesson, you will be able to:

- Describe the differences between Service Fabric, Azure Container Service and Azure Container Instances.
- Create a CI/CD pipeline using Azure Container Service, VSTS and GitHub.

Service Fabric

Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers. Service Fabric also addresses the significant challenges in developing and managing cloud native applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable, reliable, and manageable. Service Fabric represents the next-generation platform for building and managing these enterprise-class, tier-1, cloud-scale applications running in containers.



Container Orchestration

Azure Service Fabric is an orchestrator of services across a cluster of machines, with years of usage and optimization in massive scale services at Microsoft. Services can be developed in many ways, from using the Service Fabric programming models to deploying guest executables. By default, Service Fabric deploys and activates these services as processes. Processes provide the fastest activation and highest density usage of the resources in a cluster. Service Fabric can also deploy services in container images. Importantly, you can mix services in processes and services in containers in the same application.

Containers are encapsulated, individually deployable components that run as isolated instances on the same kernel to take advantage of virtualization that an operating system provides. Thus, each application and its runtime, dependencies, and system libraries run inside a container with full, private access to the container's own isolated view of operating system constructs. Along with portability, this degree of security and resource isolation is the main benefit for using containers with Service Fabric, which otherwise runs services in processes.

Service Fabric supports containers on both Linux and Windows, and also supports Hyper-V isolation mode on the latter.

ASP.NET Core in Service Fabric

In Service Fabric, one or more instances and/or replicas of your service run in a service host process, an executable file that runs your service code. You, as a service author, own the service host process and Service Fabric activates and monitors it for you.

Traditional ASP.NET (up to MVC 5) is tightly coupled to IIS through System.Web.dll. ASP.NET Core provides a separation between the web server and your web application. This allows web applications to be portable between different web servers and also allows web servers to be self-hosted, which means you can start a web server in your own process, as opposed to a process that is owned by dedicated web server software such as IIS.

In order to combine a Service Fabric service and ASP.NET, either as a guest executable or in a Reliable Service, you must be able to start ASP.NET inside your service host process. ASP.NET Core self-hosting allows you to do this.

Typically, self-hosted ASP.NET Core applications create a WebHost in an application's entry point, such as the **static void Main()** method in **Program.cs**. In this case, the lifecycle of the WebHost is bound to the lifecycle of the process.

However, the application entry point is not the right place to create a WebHost in a Reliable Service, because the application entry point is only used to register a service type with the Service Fabric runtime, so that it may create instances of that service type. The WebHost should be created in a Reliable Service itself. Within the service host process, service instances and/or replicas can go through multiple lifecycles.

A Reliable Service instance is represented by your service class deriving from StatelessService or StatefulService. The communication stack for a service is contained in an ICommunicationListener implementation in your service class. The Microsoft.ServiceFabric.Services.AspNetCore.* NuGet packages contain implementations of ICommunicationListener that start and manage the ASP.NET Core WebHost for either Kestrel or HttpSys in a Reliable Service.

Azure Container Service

Azure Container Service allows you to quickly deploy a production ready Kubernetes, DC/OS, or Docker Swarm cluster.

- Standard Docker tooling and API support
- Provisioning of DC/OS (Mesos), Docker (Swarm), and K8 (Kubernetes)
- Linux and Windows Server containers
- Deploy in the cloud or on-premises

Kubernetes

Azure Container Service for Kubernetes makes it simple to create, configure, and manage a cluster of virtual machines that are preconfigured to run containerized applications. This enables you to use your existing skills, or draw upon a large and growing body of community expertise, to deploy and manage container-based applications on Microsoft Azure.

By using Azure Container Service, you can take advantage of the enterprise-grade features of Azure, while still maintaining application portability through Kubernetes and the Docker image format.

By using these standard endpoints, you can leverage any software that is capable of talking to a Kubernetes cluster. For example, you might choose kubectl, helm, or draft.

DC/OS and Docker Swarm

DC/OS provides a distributed platform for running modern and containerized applications. With Azure Container Service, provisioning of a production ready DC/OS cluster is simple and quick. This quick start details the basic steps needed to deploy a DC/OS cluster and run basic workload.

Azure Container Service leverages the Docker container format to ensure that your application containers are fully portable. It also supports your choice of Marathon and DC/OS, Docker Swarm, or Kubernetes so that you can scale these applications to thousands of containers, or even tens of thousands.

Using Azure Container Service, you can implement a full continuous integration and deployment (CI/CD) pipeline using Azure Container Service with Docker Swarm, Azure Container Registry, and Visual Studio Team Services build and release management.

Azure Container Instances

Containers are quickly becoming the preferred way to package, deploy, and manage cloud applications. Azure Container Instances offers the fastest and simplest way to run a container in Azure, without having to provision any virtual machines and without having to adopt a higher-level service.

Azure Container Instances is a great solution for any scenario that can operate in isolated containers, including simple applications, task automation, and build jobs.

- Simplest way to run a container in Azure
  - No VMs to manage
  - Only configure the memory and CPU allocation
- Public IP Address automatically created for you
- Per-second billing based on resource consumption (CPU + Memory)
- Deploy images from popular container registries
  - Docker Hub
  - Azure Container Registry

Note: For scenarios where you need full container orchestration, including service discovery across multiple containers, automatic scaling, and coordinated application upgrades, you should use Azure Container Service instead.

Container Groups

A container group is a collection of containers that get scheduled on the same host machine. The containers in a container group share a lifecycle, local network, and storage volumes. It is similar to the concept of a pod in Kubernetes and DC/OS.

The following diagram shows an example of a container group that includes multiple containers:

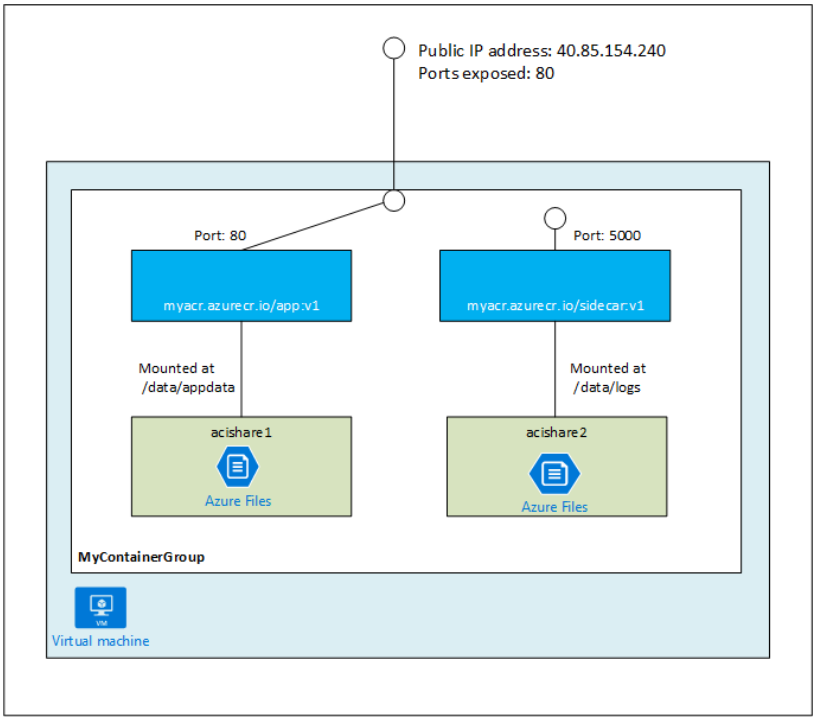


FIGURE 10.2: CONTAINER GROUP

Container groups share an IP address and a port namespace on that IP address. To enable external clients to reach a container within the group, you must expose the port on the IP address and from the container. Because containers within the group share a port namespace, port mapping is not supported. Containers within a group can reach each other via localhost on the ports that they have exposed, even if those ports are not exposed externally on the group's IP address.

You can specify external volumes to mount within a container group. You can map those volumes into specific paths within the individual containers in a group.

Multi-container groups are useful in cases where you want to divide up a single functional task into a small number of container images, which can be delivered by different teams and have separate resource requirements.

Lab: Deploying Templated Environments by Using the Cloud Shell

Scenario

Now that you have created many of the resources that you will use in your Azure application, you have decided to automate the creation of your assets in Azure. Some of your administrators are Windows experts and would prefer to automate using PowerShell while others use Linux and would prefer to automate from the command line. Due to this requirement, you will try and implement automation using PowerShell and separately using the CLI interface.

Objectives

After you complete this lab, you will be able to:

- Launch the Cloud Shell.
- Deploy an ARM template using the Cloud Shell.

Lab Setup

Estimated Time: 60 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Start the Cloud Shell

Exercise 2: Use the Cloud Shell to Deploy an ARM Template

Module Review and Takeaways

In this module, you explored how you can deploy modern applications using ARM Templates or popular container formats. Using these methods, you can deploy exact replicas of your solutions across environments, geographies or even subscriptions.

Module 11

Securing Azure Web Applications

Contents:

Module Overview

Just like on-premises applications, applications in the cloud need streamlined security mechanisms that are flexible. Azure Active Directory is an identity provider that can provide identity and access functionality for your custom applications or SaaS applications. Lesson 1, "Azure Active Directory," introduces the Azure AD service. Lesson 2, "Azure AD Directories," details how to create a directory in Azure AD. Lesson 3, "Azure AD Offerings," describes the various offerings available in Azure AD such as B2B, B2C, and multi-factor authentication. Lesson 4, "Azure Key Vault," introduces the Azure Key Vault service designed to manage secrets for workloads and applications.

Objectives

After completing this module, you will be able to:

- Describe the Azure AD service.
- Explain the features that are available for the directories in Azure AD.
- Describe the Microsoft Azure Multi-Factor Authentication service.

Lesson 1
Azure Active Directory

Azure AD provides a suite of services that you can integrate with custom applications, on-premises machines, existing domains, and third-party services.

This lesson describes the Azure AD service and its features and benefits.

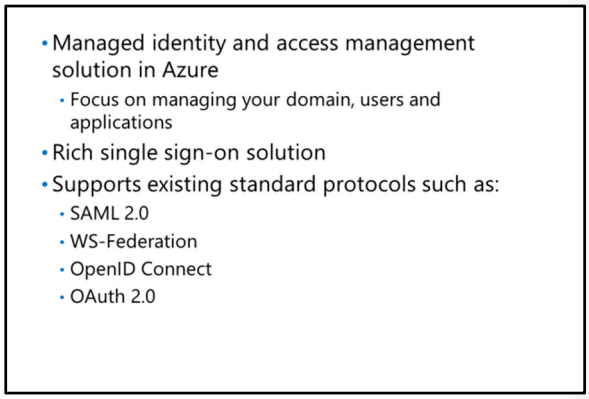
Lesson Objectives

After completing this lesson, you will be able to:

- Explain the benefits of Active Directory in Azure.
- List the Active Directory services in Azure.

Azure Active Directory Overview

Azure Active Directory (Azure AD) is a service that offers the identity and access capabilities of Active Directory for use with your applications whether they are on premise or hosted in the cloud. Azure AD can be used to:



- Implement single sign-on (SSO) and sign-out for your custom line of business (LOB) applications and various third-party software as a service (SaaS) providers.
- Query and modify directory objects including users, applications, and groups by using a standard API.
- Integrate your cloud applications and SaaS applications with your existing on-premises identity management systems by syncing identities and optionally credentials.

With identity sync, your existing corporate credentials can be used to authenticate to new or existing applications that are hosted in Azure. These credentials can also be used with third-party SaaS applications such as Dropbox, Intuit, or Skype. Azure AD also offers a self-service portal where your users can optionally manage their own passwords or groups. By using the password write-back feature, the updated password hash is then duplicated back to your on-premises Active Directory instance.

Application developers can use Azure AD as an identity provider in their custom applications to provide a true SSO experience to users. An existing application can be updated to use a specific Azure AD tenant for identity. Your SaaS applications can also be modified to support Azure AD as an identity provider.

Azure AD is already in use by many cloud services today, such as Microsoft Intune and Office 365. These services rely on the identity management capabilities provided by Azure AD. These capabilities include a cloud-based store for directory data and a core set of identity services including user logon processes and authentication and federation services.

Relationship between Active Directory and Azure AD

Similar to how Active Directory serves as the data store for identities in your on-premises environment, Azure AD provides a repository for all of your organization's directory data in the cloud, so that it can be readily available to all the services you have subscribed to. Similar to how an LOB application might use Lightweight Directory Access Protocol (LDAP) to access data in your local Active Directory, third-party cloud applications can interact with your data in Azure AD by using the Graph API.

Local or cloud applications use a similar methodology to access identity data stored in a directory:

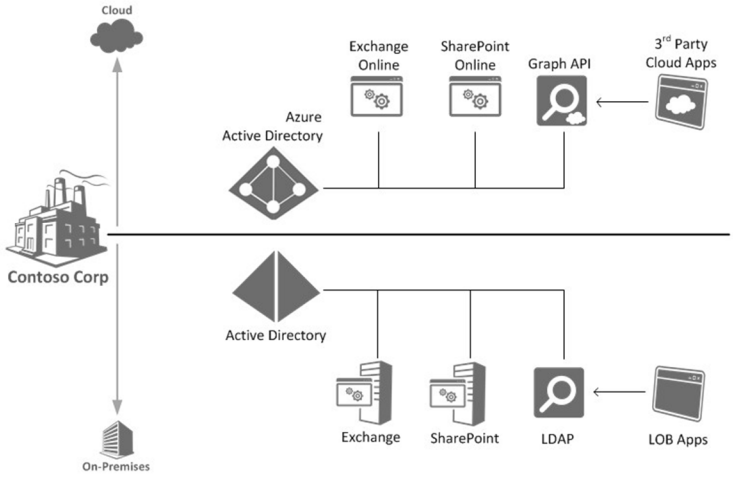
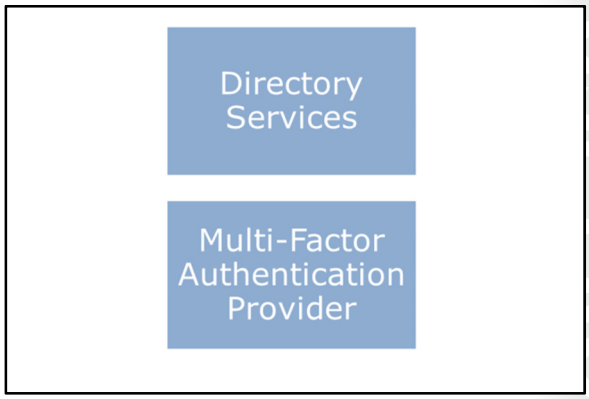


FIGURE 11.1: ACTIVE DIRECTORY AND AZURE AD

Azure AD Services

Azure AD is composed of multiple features. This module focuses on two of these features, directories and Multi-Factor Authentication.



Directory Services

Azure AD provides conceptual directories where you can store related user accounts. Directories can store identities synced from on premises systems, identities created in Azure, and third-party identities. These identities can then be configured for use with SaaS applications.

Multi-Factor Authentication

Multi-Factor Authentication offers a second layer of authentication for your applications that is completely managed. Your administrators simply need to configure Multi-Factor Authentication and your applications can take advantage of the feature by using Azure AD as the authentication (identity) provider. Multi-Factor Authentication supports authentication from mobile apps, text messages, or phone calls.

Access Control Service

 **Note:** ACS is a deprecated service. You might inherit applications that use this service because the service is still operational.

Access Control service (ACS) is a service in Azure that federates multiple identity providers to a single set of standardized claims. Normally you would need to write code for each identity provider and handle their claims in a custom manner. ACS allows you to add identity providers that implement OAuth 2.0 and map their claims to a new set of claims. For example, you can use ACS to map the claims from Microsoft, Google, Yahoo, and Facebook to a single set of claims that your application can easily expect. This greatly simplifies the amount of code that is necessary to support multiple identity providers. ACS also supports identity providers that use WS-Trust or WS-Federation as their protocols. ACS can also optionally host a logon page for your application.

ASP.NET identity provides similar functionality and is largely used in scenarios that were previously appropriate for ACS.

Lesson 2
Azure AD Directories

Azure AD directories provide a logical way to group your users and applications.

This lesson introduces directories and details the different components and integrations available for Azure AD.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how users and third-party accounts can be added to a directory.
- Describe the integration of SaaS applications and Azure directories.
- Describe the Azure AD Graph API.

Managing Directories

Directories provide a simple and logical way to group related identities. A directory can consist of the following three types of identities:

- Users synced from existing Active Directory installations (on-premise identities)
- Users added manually to the directory (cloud-only identities)
- Third-party accounts (third-party identities)

You can manage directories from a variety of locations. This list of locations is not intended to be exhaustive.

- You can manage your organization's tenant data using any of these tools:
  - Microsoft Azure AD Portal
  - Windows Intune Account Portal
  - Microsoft Azure Management Portal
  - Office 365 Account Portal
- In the Management Portal, you can perform tasks such as:
  - Create, modify, and dispose user accounts
  - Manage passwords

The various management experiences for Azure AD can all be used together:

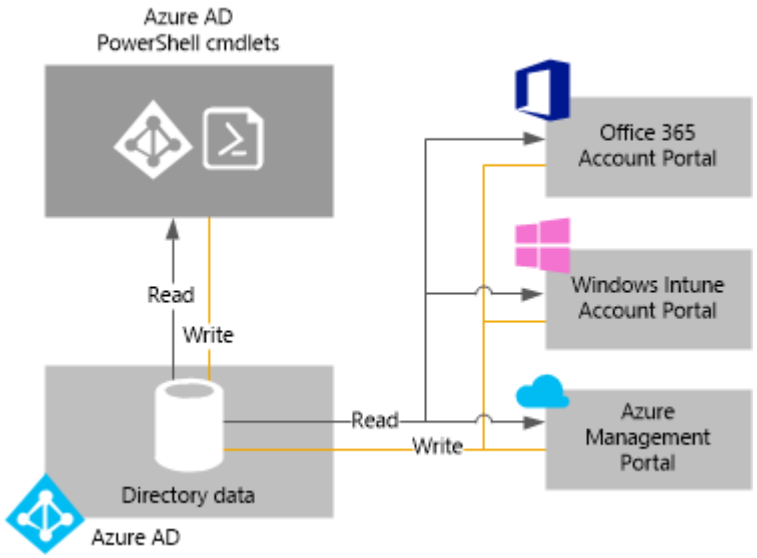


FIGURE 11.2: AZURE AD MANAGEMENT

Microsoft Intune Account Portal and Office 365 Admin Center

You can use an account portal to manage your Office 365 or Microsoft Intune subscription and specify the users who can access its various services. From the account portal, you can perform tasks such as manually adding user accounts and security groups, setting up and managing service settings, checking service status, and accessing online help. Users can also access these account portals but only to change their password or to access the various services for which they have been assigned licenses.

Microsoft Azure Management Portal

You can use the Azure Management Portal to manage the services associated with your Azure subscription. If you have an existing Azure subscription that is using your Microsoft account, you also can use the Management Portal to manage your directory. Most Azure subscriptions include a default Azure AD instance. When you sign up for Azure as an organization, a directory tenant is automatically created for you based on the value you provide in the Organization Name field during sign-up.

Windows PowerShell

You can use the Microsoft Azure Active Directory Module for Windows PowerShell cmdlets to accomplish many Azure AD tenant-wide administrative tasks. Administrative tasks, such as user management and domain management, and configuring SSO can be automated by using Windows PowerShell scripts or by using a service such as Azure Automation.

Integrating On-Premises Directories with Azure AD

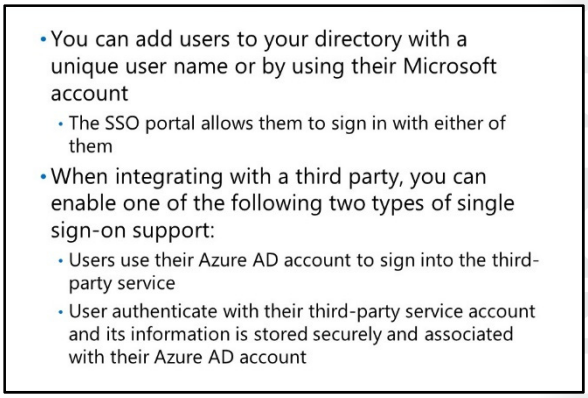
If your organization has an on-premises directory service, you can integrate it with your Azure AD directory. One of the primary benefits of setting up directory integration capabilities such as directory sync or SSO, is that after you've configured the sync operation, all the cloud services that you have subscribed to in your Azure AD tenant can utilize the data that is now provisioned and updated in your cloud store. Various sync options are available including:

- Syncing identity from Active Directory to Azure AD.
- Syncing identity and password hash from Active Directory to Azure AD.
- Syncing identity and password hash from Active Directory to Azure AD and enabling password writeback.

Writeback is a feature that allows your existing Active Directory identity to be updated when a change occurs in Azure AD. When you sync identity and password hash, it is important to remember that you are creating two identities. Managing these identities and their sync relationship is a key part of designing authentication schemes for cloud applications.

Directory Users

You can directly create accounts in a directory for every user who accesses your services. You also can manage the accounts or delete them when they are no longer needed. By default, users do not have administrator permissions, but you can optionally assign permissions to them. There are three types of users that you can create by using the Management Portal:



- New user in your directory or organization
- User with an existing Microsoft account
- User in another Azure AD directory

You can create new users in the portal by providing the following details about the user:

- First Name
- Last Name
- Display Name
- Alias
- Role

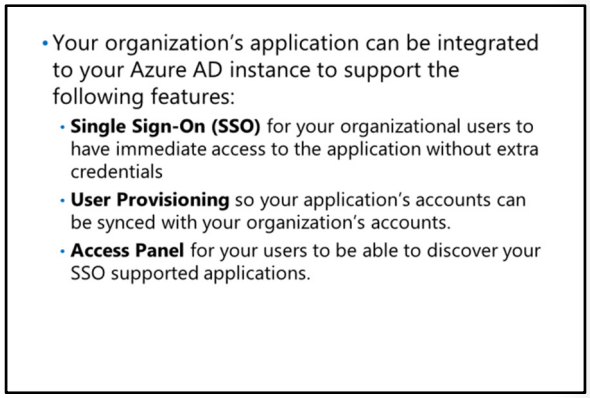
After you create a new user, a temporary password is generated. You can then email this password to the user. On the first login, the user will be prompted to change the temporary password.

External Users

To an Azure AD directory, you can add users from another Azure AD directory or users with Microsoft accounts. This enables the external users to collaborate with users who already exist in your directory. This is useful for collaborating in an environment with users who need to manage directory resources, such as applications, without requiring those users to have an account and credentials in your directory.

When you add a user from one directory into a new directory, that user is an external user in the new directory. Initially, the display name and user name are copied from the user's home directory and stamped onto the external user in the other directory. From then on, the profile properties of the external user object are entirely independent. If you make a change to the user in the home directory, such as changing the user's name, adding a job title, and so on, those changes are not propagated to the external user account in the other directory.

Applications in Azure AD



Enterprise developers and SaaS providers can develop commercial cloud services or LOB applications that can be integrated with Azure AD to provide secure sign-in and authorization for their services. Azure AD also includes an access panel for users where they can discover what applications they can access. From this panel, they can access their applications by using SSO. To integrate an application or a service with Azure AD, a developer must first register the details about the application with Azure AD by using the Management Portal. These steps are similar to the steps for adding an SSO third-party application to your Azure AD instance.

Single Sign-On

Configuring SSO enables the users in your organization to be automatically signed in to any third-party SaaS application using their Azure AD credentials. This functionality provides users with the convenience of remembering a single password and it also increases the organization's security by providing users with access to only their applications. Azure AD can federate its identity to your custom application, store the custom application's credentials, or integrate with a third-party SSO provider.

User Provisioning

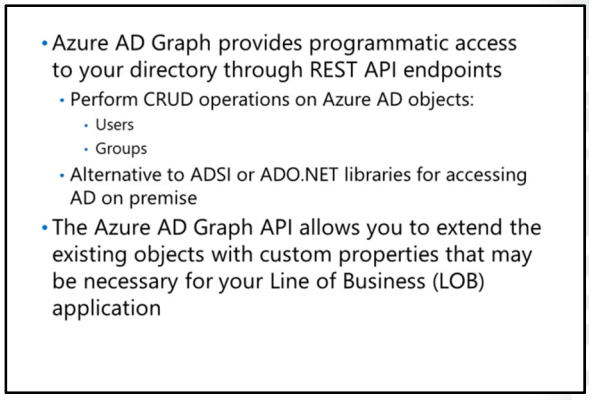
User provisioning enables automated user provisioning and deprovisioning of accounts in third-party SaaS applications from within the Management Portal by using your Windows Server Active Directory or Azure AD identity information. When a user is given permissions in Azure AD for one of these applications, an account can be automatically created (provisioned) in the target SaaS application. When a user is deleted or his or her information changes in Azure AD, these changes are also reflected in the SaaS application. User Provisioning allows your application to automate identity lifecycle management and enables administrators to control and provide automated provisioning and de-provisioning of user accounts from SaaS applications.

Access Panel

The access panel in Azure AD offers a single dashboard for your organization. Users can access one or more applications that you manage from within the Azure AD instance using a single sign-on experience directly from this panel. Users do not require an Azure or Office 365 subscription to connect to the access panel.

Azure AD Graph

The Graph API provides programmatic access to Azure AD through REST API endpoints. This API can be used to store and retrieve metadata about your users that is not part of the typical user profile in Active Directory.



Create, Read, Update and Delete (CURD) Operations

Applications use the Graph API to perform CRUD operations on directory objects in your Azure AD instance. For example, you can use the Graph API to perform the following operations on a user object:

- Create a new user in a specified directory.
- Get detailed properties for a user.
- Check group membership for a user.
- Update the extended properties (profile) of a user.
- Disable or delete a user account.

In addition to users, similar operations are supported for groups and applications in Azure AD. To call the Graph API on a specific directory, you must register the application with Azure AD and configure it to allow access to the directory.

Directory Extensions

Many applications require metadata and properties for each user that is not typically stored in a standard Active Directory user profile. The Graph API allows you to register and then use extended properties. For example, if you need to store and then retrieve the Xbox Live ID for each user in a gaming social application, you must first register the new property in the directory. You can then use this property in subsequent operations because it is not available for every user object in the directory.

Lesson 3
Azure AD Offerings

Multi-Factor Authentication is a feature in Azure AD that you can use to provide an additional layer of authorization to your existing directory accounts. This authorization could be a phone call, mobile code or custom application. Azure AD B2B and B2C are services in Azure AD that can extend the reach of your directory to include partner businesses and customers.

This lesson will introduce and dive into various offerings in Azure AD.

Lesson Objectives

After completing this lesson, you will be able to:

- Detail the differences between Azure AD B2B and B2C.
- Describe multi-factor authentication in Azure AD.
- List the multi-factor authentication providers that are available for Azure AD.

Azure AD B2B

Azure AD business-to-business (B2B) collaboration capabilities enable any organization using Azure AD to work safely and securely with users from any other organization, small or large. Those organizations can be with Azure AD or without, or even with an IT organization or without.

- Provides access to your organization's data and applications
  - Specifically targeted partner organizations and collaborators
  - Designed for partner users acting "on behalf of" their organization
- Governs access to host applications and resources
  - Access reviews
  - Email verification
  - Allowlist/denylist
- Partner users are discoverable within the directory

Organizations using Azure AD can provide access to documents, resources, and applications to their partners, while maintaining complete control over their own corporate data. Developers can use the Azure AD business-to-business APIs to write applications that bring two organizations together in more securely.

Azure AD B2C

Azure AD B2C is a cloud identity management solution for your web and mobile applications. It is a highly available global service that scales to hundreds of millions of identities. Built on an enterprise-grade secure platform, Azure AD B2C keeps your applications, your business, and your customers protected.

- A consumer identity and access management service
- Integrates across your modern platforms
- Highly available and scales to hundreds of millions of consumers
- Supports a customizable experience for consumers

With minimal configuration, Azure AD B2C enables your application to authenticate:

- **Social Accounts** (such as Facebook, Google, LinkedIn, and more)
- **Enterprise Accounts** (using open standard protocols, OpenID Connect or SAML)
- **Local Accounts** (email address and password, or username and password)

Policies

The extensible policy framework of Azure Active Directory (Azure AD) B2C is the core strength of the service. Policies fully describe consumer identity experiences such as sign-up, sign-in, or profile editing. For instance, a sign-up policy allows you to control behaviors by configuring the following settings:

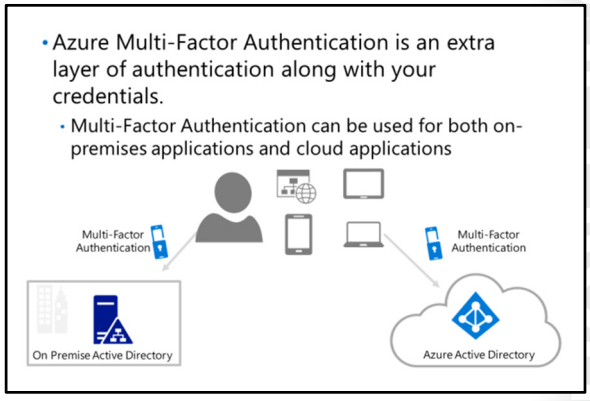
- Account types (social accounts such as Facebook or local accounts such as email addresses) that consumers can use to sign up for the application
- Attributes (for example, first name, postal code, and shoe size) to be collected from the consumer during sign-up
- Use of Azure Multi-Factor Authentication
- The look and feel of all sign-up pages
- Information (which manifests as claims in a token) that the application receives when the policy run finishes

You can create multiple policies of different types in your tenant and use them in your applications as needed. Policies can be reused across applications. This flexibility enables developers to define and modify consumer identity experiences with minimal or no changes to their code.

Policies are available for use via a simple developer interface. Your application triggers a policy by using a standard HTTP authentication request (passing a policy parameter in the request) and receives a customized token as response.

Multi-Factor Authentication

Multi-factor authentication is an additional layer of security that can protect applications from unauthorized access if a user's credentials are compromised. To the end user, they simply provide additional means of authentication that can include things such as a phone, RSA key, or custom device. Multi-factor authentication is usually defined by having the user provide two things:



- Something you know: password
- Something you have: trusted device (phone, smartwatch, and so on)

Multi-factor authentication's strength is in its multiple layer approach. If a user's credentials are compromised, a malicious user would still require a trusted device that is assigned to the same user to compromise the application or its data. Typically, if a user loses a trusted device, they report it immediately and the device can be de-authorized.

Out of the box, Azure AD uses passwords as the default credential for user access. Multi-Factor Authentication is a service in Azure AD that implements the previously mentioned multi-factor authentication pattern. You can use multi-factor authentication with either Azure AD or an on-premises directory. The second form of authentication can be a smartphone, a phone number that supports calls or text messages, or a custom application. When using the Multi-Factor Authentication service with Azure AD, administrators can enable multi-factor authentication specifically for each individual user. The Multi-Factor Authentication service supports up to three phone numbers that are authorized for use as a second form of authentication. The user can also opt to use the multi-factor authentication mobile apps that support both push notifications and one-time pass codes as authentication options.

Multi-Factor Authentication can be enabled for cloud or on-premises applications:

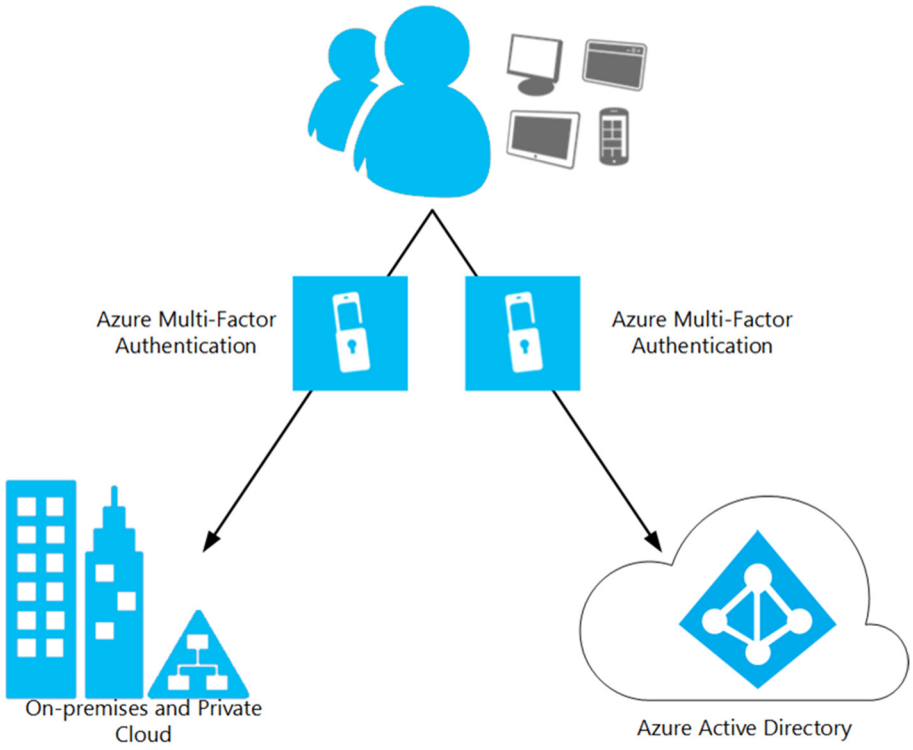


FIGURE 11.3: AZURE MFA

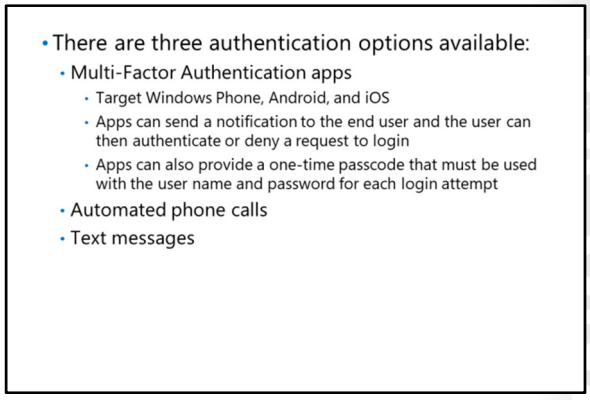
Reference Link: https://docs.microsoft.com/azure/multi-factor-authentication/multi-factor-authentication-get-started-server

A software development kit (SDK) is available to integrate your custom applications with Azure AD Multi-Factor Authentication. The SDK allows you to use the Multi-Factor Authentication phone call or text message verification options as part of your custom application's sign-in process. This is useful if you are building a custom application that does not redirect to Azure AD's sign-in page and instead has a built-in logon form.

Reference Link: https://docs.microsoft.com/azure/multi-factor-authentication/multi-factor-authentication-sdk

Multi-Factor Authentication Providers

You can use the Multi-Factor Authentication (MFA) service in conjunction with Windows Server Active Directory Domain Services (AD DS) or Azure AD to help secure both cloud and on-premises applications. Users in your organization have many different options available for their second form of authentication with your application when signing in using Azure AD. As an administrator, you can control which options are available to end users.



Multi-Factor Authentication Apps

Apps are already available in the individual app stores for Windows Phone, Android, and iOS to integrate with Azure AD Multi-Factor Authentication. Users download these apps and activate them by using a setup code. When the user signs in to your applications, a notification is pushed to the app on their mobile device. The user can then immediately approve or deny the authentication. Azure AD Multi-Factor Authentication also uses an open standard for authentication and can be used with a variety of third-party multi-factor authentication applications. These applications generate a one-time use pass code that the user must enter after they attempt to log on. This behavior is similar to an RSA-key device.

Automated Phone Calls and Text Messages

Users have the option to have an automatic phone call or text message placed to their authorized mobile device. For the phone call, the user only has to answer the call and press the pound (#) key on his or her phone to complete the sign-in process. For the text messages, users are sent a one-time use pass code that they must enter after they attempt to log on.

Lesson 4
Azure Key Vault

Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. By using Key Vault, you can encrypt keys and secrets (such as authentication keys, storage account keys, data encryption keys, .PFX files, and passwords) using keys protected by hardware security modules (HSMs).

This lesson introduces the Azure Key Vault service and describes how to store secrets in the service.

Lesson Objectives

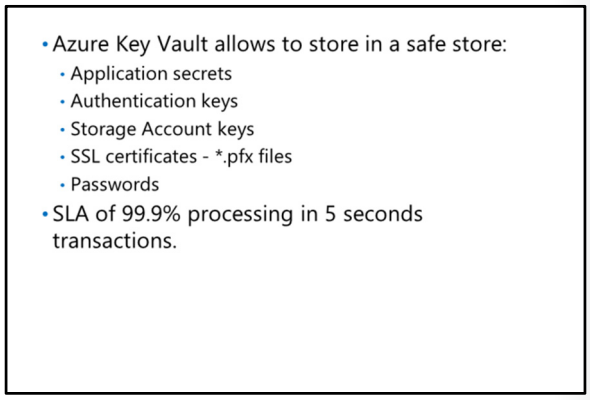
After completing this lesson, you will be able to:

- Describe the Azure Key Vault service.
- Use PowerShell to manage keys in Azure Key Vault.

Azure Key Vault

Every day, more and more applications are created and deployed in the cloud. All those applications need to store application secrets, keys, passwords ant/or other secrets in a safe area. Azure Key vault is the service to keep these secrets away from non-authorized access. This service allows you to store:

- Application secrets
- Authentication keys
- Storage Account keys
- SSL certificates - *.pfx files
- Passwords

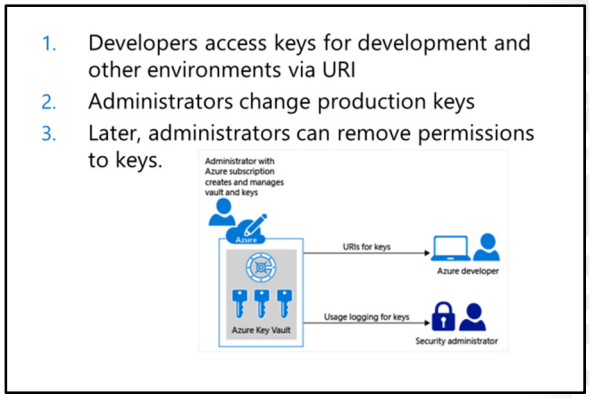


The Azure Key vault service provides a state of the art service to secure your secrets but also achieve this with High-performance. Microsoft has an SLA of 99.9 % with a secret processing time of 5 second transactions.

Utilization

Using the Azure Key vault service, your development team can use cryptographic keys, certificates, passwords, and other applications secrets without the need to access them. The Key Vault service allows a client application or service to access data using a URI without storing the data locally on their side. This will improve security of your application. The general steps will be:

1. Developers access keys for development and other environments via URI.
2. Administrators change production keys.
3. Later, administrators can remove permissions to keys.



Note: Using Azure Key Vault your developer team will never have access to the keys but will be able to access the resources they need.

Lab: Integrating Azure Active Directory with the Events Administration Portal

Scenario

Even though the Contoso Events web application is public, the Administration application should be locked down to users only from your domain. You have decided to use Azure AD and ASP.NET identity to provide this functionality. In this lab, you will create a new ASP.NET project by using the ASP.NET identity framework and integrate the project with Azure AD. The website will then use your organization accounts for signing in.

Objectives

After you complete this lab, you will be able to:

- Create an Azure AD by using the Management Portal.
- Create users in Azure AD by using the Management Portal.
- Create a new MVC project that uses Azure AD organizational accounts for security.

Lab Setup

Estimated Time: 60 minutes

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Creating an Azure AD Directory

Exercise 2: Securing an Existing ASP.NET Web Application

Exercise 3: Integrating Azure AD with ASP.NET Identity

Question: What other identity providers could you use with ASP.NET Identity?

Module Review and Takeaways

In this module, Azure AD is presented as a solution for identity management in many different scenarios. With the ACS offering, the third-party SaaS integration, and Multi-Factor Authentication, Azure AD provides unique services that you can use to improve your existing on-premises identity solution.

Note: The ASP.NET Identity framework is a newer way of securing web applications, and it has certain advantages over Membership and Forms Authentication.

Review Question

Question: When you use Azure AD ACS, why should you remap the claims from each identity provider?

Course Evaluation

Your evaluation of this course will help Microsoft understand the quality of your learning experience.

Please work with your training provider to access the course evaluation form.

Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.