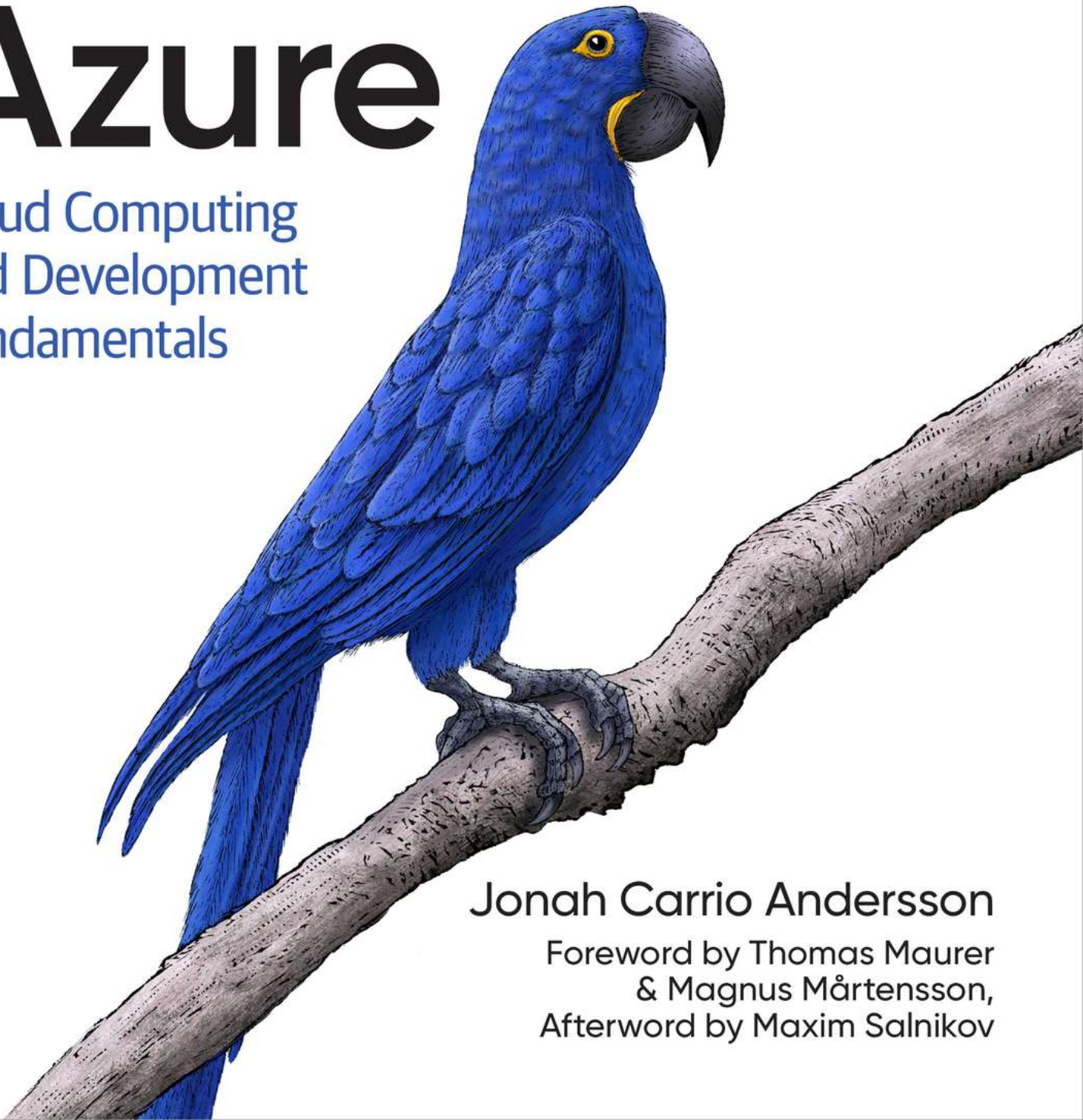


O'REILLY®

Learning Microsoft Azure

Cloud Computing
and Development
Fundamentals



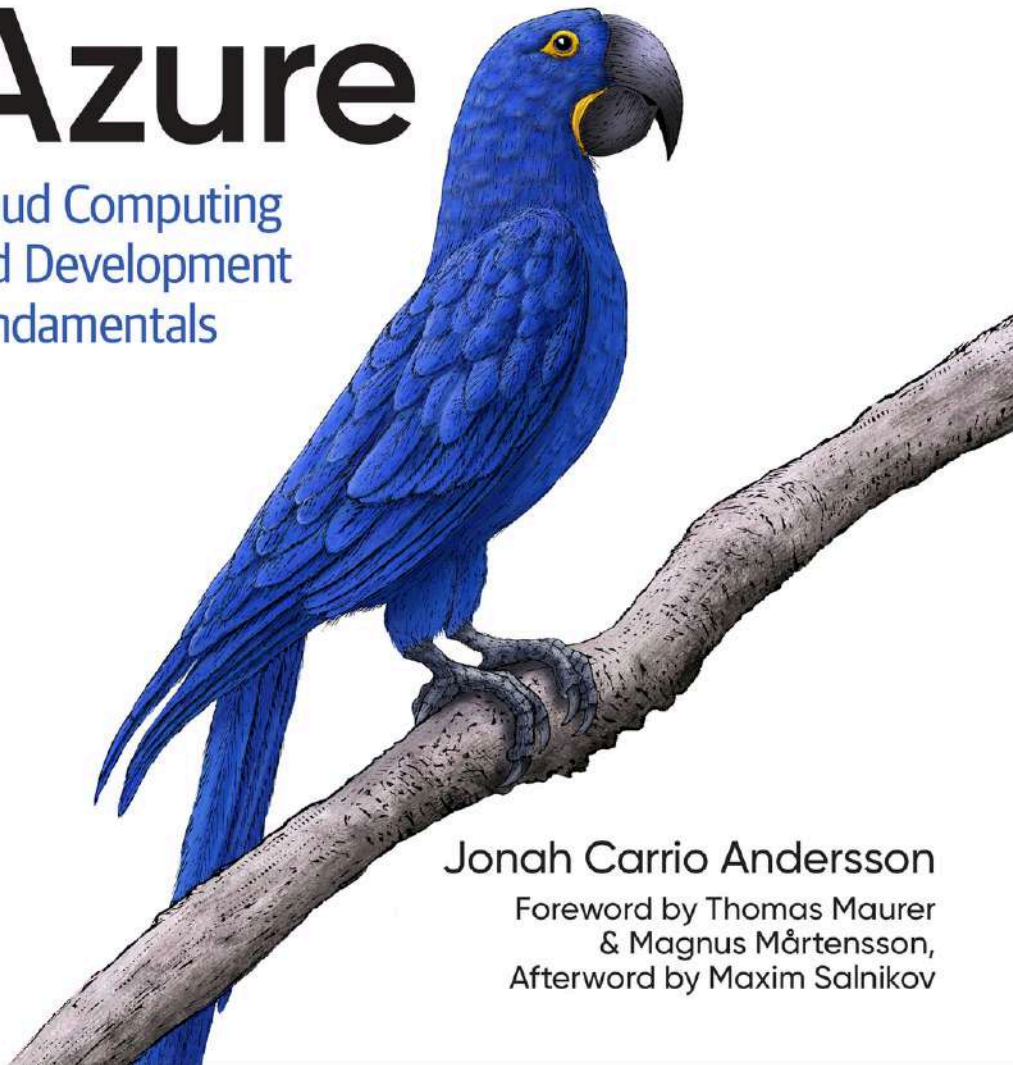
Jonah Carrio Andersson

Foreword by Thomas Maurer
& Magnus Mårtensson,
Afterword by Maxim Salnikov

O'REILLY®

Learning Microsoft Azure

Cloud Computing
and Development
Fundamentals



Jonah Carrio Andersson

Foreword by Thomas Maurer
& Magnus Mårtensson,
Afterword by Maxim Salnikov

Praise for Learning Microsoft Azure

Learning Microsoft Azure is a great resource for anyone seeking to advance their knowledge in the vast ecosystem of Azure. This book navigates the entire spectrum of Azure services, catering equally to novices and veterans in the field.

Jonah's meticulous approach in detailing Azure's components mixes architectural insights with practical examples, deepening the reader's understanding. The inclusion of skill assessments ensures that learners are not only consuming information but actively engaging with it.

What sets this book apart is its coherent explanation of how the many Azure services connect, demystifying complex terminology and providing guidance on service utilization. It's clear that Jonah has a passion for making others grow, anyone who has met Jonah will feel that passion while reading the book.

I recommend Learning Microsoft Azure to anyone aspiring to broaden their Azure skills. It's not just a book; it's a comprehensive guide that prepares you for the future of cloud computing.

—Mathias Olausson, CTO at Solidify, Microsoft MVP

This book is a fantastic journey into Azure, perfect for beginners but also a goldmine for experts. It's the kind of all-around guide that lightens the path for the uninitiated and also serves as a handy reference for seasoned professionals.

—Peter T. Lee, Managing Delivery Architect, Capgemini
America, Inc.

Azure is such a vast platform that it may seem only an expert could comprehend its many components. In Learning Microsoft Azure, Jonah Andersson masterfully relates the entire stack for a general technical audience. The book is carefully constructed yet easy to read, delivering crucial knowledge about Azure and cloud computing more generally throughout. Data professionals like me will particularly benefit from understanding Azure's machine learning and AI capabilities. I recommend Learning Microsoft Azure to any tech professional seeking to gain a hands-on, technical understanding of cloud computing through Azure.

—George Mount, Data analyst and educator at Stringfest Analytics

This is the book I wish I had when I started my journey with Microsoft Azure. Jonah has a fantastic ability to simplify complex technical concepts, making them easy to understand. Regardless of whether you are a beginner or a seasoned Cloud Engineer, this book will help you understand and learn Microsoft Azure to the fullest, enabling you to benefit from its full range of features and capabilities.

—Nikolaos Delis, Senior Cloud Engineer and Cloud Evangelist, Helo AB

In Learning Microsoft Azure, Jonah delivers a comprehensive and accessible book for those working in cloud computing. With clear explanations, this book is a great starting point for beginners and also serves as a handy reference for experienced professionals. It is an indispensable asset that presents insightful use cases and breaks down complex concepts so they can be easily understood regardless of how much experience the reader has. Whether you're just getting started or aiming for advanced knowledge, this book is a valuable resource for understanding the basics of Azure.

—Priyanshu Jimish, Technical Evangelist, Virta Technologies

Jonah Andersson's Learning Microsoft Azure thoroughly covers the core facets of Azure but also delves deeply into the nuances of integration, DevOps, and governance. An indispensable resource for any engineer aiming for comprehensive mastery over Azure.

—Dr. Milan Milanović, CTO and Microsoft MVP

Writing a review for a book is never an easy task, especially when the book is as comprehensive and useful as this book. Learning Microsoft Azure contains practical information on Azure Cloud services, Azure DevOps pipelines, Azure networking, containerization, AI and more.

This book demonstrates Jonah Andersson's superior training skills and knowledge on cloud computing. It provides you with broad, complete and practical information to implement your solutions in Microsoft Azure.

—Reza Salehi, Consultant, Author of Azure Cookbook

It doesn't matter if you are already experienced or just starting to learn Azure and want to gain a deeper understanding. This book pulls everything you need together and is thoughtfully organized, making it easy to learn.

—Lior Yantovski, DevOps Tech Lead, AT&T

This book is an incredible resource for those who want to learn Microsoft Azure and cloud concepts, spanning from beginner to advanced level. Reading this book will help you implement cloud solutions in real time projects. Jonah Andersson endeavored to cover each and every topic with detailed information and an accessible way of writing.

—Sagar Rastogi, Technical Architect, Tata Consultancy
Services

Learning Microsoft Azure by Jonah Carrio Andersson is an exceptional guide that offers a comprehensive and insightful journey through the intricate world of Microsoft Azure. Andersson's expertise shines through, providing readers with a well-structured and detailed exploration of cloud computing, Azure services, AI, IoT, and DevSecOps, making it an indispensable resource for anyone looking to master Azure technologies. Her clear explanations and practical insights make this book a valuable asset for both beginners and experienced professionals seeking to harness the power of Azure for their projects and career development.

—Amit Dass, Data Architecture Manager, MCT Certified,
Microsoft Azure Data Architect and Lead Data Engineer,
Certified Professional Data Engineer and Architect in
GCP, Azure, and AWS (Multi-cloud)

A very well-written, well-researched and engaging book on Azure. The knowledge and passion of the author really shines through!

Learning Microsoft Azure works well for people with technical and non-technical backgrounds alike. It will also help anyone who wants to achieve Azure certifications. Highly recommended to anyone looking to develop a solid foundation in Azure.

—Andrew Urwin, Microsoft Azure MVP and Director of
Platform and DX at Clue Software

Jonah has not only captured and eloquently explained the concepts pertaining to cloud and Azure fundamentals, but she has brought together the building blocks needed for developers to create apps and deploy services on Azure. She has captured what is necessary to get a start on DevOps on Azure that can translate to other cloud providers as well. Her passion for teaching others shines in this text!

—Dwayne Natwick, ISC2, Microsoft, and CompTIA
trainer and consultant; CEO, Captain Hyper-scaler, LLC

Jonah Andersson's book isn't just a guide; it's a treasure map to unlock the full potential of Microsoft Azure. With its comprehensive insights into cloud migration, development, and deployment, it's a must-read for adventurers in the Azure realm.

—Saeid Dahl, Azure Tech Leader, Sr. Solution Architect at WeSafe, and Microsoft Learning Expert

The Azure platform can be daunting for beginners, however, in this book, Jonah has managed to make learning the fundamentals easy and understandable. By enriching the technical details with many practical and personal experiences she has brought the examples to life and helped us see how they can be used in the real world. The breadth of her knowledge makes this book accessible to both people just starting out and to those who are more experienced and looking to expand their knowledge.

—John Kilmister, Software Architect and Microsoft Azure MVP

Jonah Andersson's book is an essential read for any IT professional looking to harness the full potential of cloud computing. With its comprehensive coverage, practical examples, and expert insights, this book elevates the reader's understanding of Azure to new heights, covering topics like cloud-native development, DevOps, migration strategies, and much more!

—Freek Berson, Principal Product Manager at Parallels (Alludo) and Microsoft MVP

In Learning Microsoft Azure, Jonah Andersson brilliantly demystifies Microsoft's Azure platform, unveiling its full potential. A must-read guide for unlocking the limitless possibilities of cloud technology.

—Gerald Versluis, Senior Software Engineer at Microsoft

Jonah Andersson's book, Learning Microsoft Azure, is a beacon in the vast and ever-expanding landscape of Azure. This book skillfully guides beginners through important cloud concepts, making the immense world of Azure accessible and navigable.

—Michael John Peña, Technical Director at Playtime
Solutions and Microsoft Azure MVP

When learning about the cloud, it can be overwhelming as there are so many services, design patterns, and technical best practices that vary between workloads. Jonah has done an excellent job breaking down the complex facets of Microsoft Azure, from storage to compute and everything in between. I would recommend this book as the “go-to” reference guide for anyone deploying Azure services and building out performant cloud architectures.

—Colby T. Ford, Ph.D., Founder of Tuple—The Cloud
Genomics Company, Microsoft MVP (Azure), and Author
of Genomics in the Azure Cloud

Jonah guides the reader through a clear narrative to learn about Microsoft Azure, from the fundamental concepts of service models and the core concepts of Azure, through to compute, databases, AI services, and governance. This is the book that all those looking to begin their Azure cloud journey, and to get a breadth of knowledge, should be reaching for.

—Jacqui Read, software architect and Author of
Communication Patterns: A Guide for Developers and
Architects

Jonah Andersson's book, Learning Microsoft Azure, of which I had the honor to be one of the technical reviewers, is an outstanding handbook for anyone who wants to get involved or even just comprehend what cloud computing is, understand specifically what Microsoft Azure is, and navigate what it can offer. Jonah's expertise is apparent in every chapter as she helps the reader learn about each service offered while sharing her experience.

—George Chrysovalantis Grammatikos, Microsoft Azure
MVP and Azure Solutions Architect, Tiski Ltd.

Jonah Andersson's Learning Microsoft Azure is a masterful guide that transforms complexity into clarity, presenting the ins and outs of cloud infrastructure with precision and practicality. A must-read for anyone aspiring to excel in the cloud computing sphere.

—Lars Klint, Principal Advocate, Arkahna and Microsoft
Azure MVP

Learning Microsoft Azure is a comprehensive guide for beginners that want to start embracing the Azure cloud, but also a great guide for cloud experts and solution architects that want to improve their knowledge on this platform. It's a nice read that explains all the different services, from infrastructure to development. I recommend it to everyone that loves the Microsoft cloud.

—Stefano Demiliani, Microsoft MVP—Azure and
Business Applications, CTO at EID NAV-lab Group

Jonah Andersson is a dynamic community advocacy leader, continually inspiring both current and future community members with her unwavering passion and dedication.

Her book Learning Microsoft Azure expertly weaves a comprehensive overview, delivering Azure's fundamentals with precision and depth, making it an invaluable technical companion.

—Kevin Evans, a.k.a the NETRUNNER, Cloud Solution
Architect, Microsoft

Jonah Andersson has done a fantastic job of distilling complex technical topics into an approachable, easy to read book for technical professionals of all levels. Learning Microsoft Azure will provide readers with an opportunity to learn critical cloud computing concepts as well as develop fundamental hands-on skills with Microsoft Azure.

—Dave McCollough, Chief Technology Officer, Geokey,
Inc.

Learning Microsoft Azure is an exceptionally well-structured and comprehensive guide that provides a fantastic introduction to the diverse functionalities of the Azure platform. As a freelance developer I see a lot of different tech stacks and that's why I really appreciate good sources of information that I can lean on in my work. This book really hits the spot for me. It is well-written, well organized and the author's reputation for being incredibly supportive and empowering to fellow developers really shines through. Highly recommended for those seeking to unlock their potential in Azure!

—Steffen Jørgensen, IT Consultant at Sigma Solid

Learning Microsoft Azure

Cloud Computing
and Development Fundamentals

Jonah Carrio Andersson



Beijing • Boston • Farnham • Sebastopol • Tokyo

Learning Microsoft Azure

by Jonah Carrio Andersson

Copyright © 2024 Jonah Carrio Andersson. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

- Acquisition Editor: Megan Laddusaw
- Development Editor: Jill Leonard
- Production Editor: Aleeya Rahman
- Copyeditor: Piper Editorial Consulting, LLC
- Proofreader: Charles Roumeliotis
- Indexer: nSight, Inc.
- Interior Designer: David Futato
- Cover Designer: Karen Montgomery
- Illustrator: Kate Dullea
- December 2023: First Edition

Revision History for the First Edition

- 2023-11-17: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781098113322> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Learning Microsoft Azure*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

The views expressed in this work are those of the author, and do not represent the publisher's views. While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Microsoft. See our [statement of editorial independence](#).

978-1-098-11332-2

[LSI]

Dedication

I would like to dedicate this first book I have ever written to my late and beloved mother, Joan, who died of breast cancer in 2004. Back then at the age of 18, I learned to take my journey in life with resiliency and courage thanks to my mother, who placed her trust and confidence in me. I hope she sees how far I've reached in this life and what I became because of her inspiration and motivation.

I also want to dedicate this book to my family in the Philippines. I devoted most of my life to taking care of my younger siblings and family, and they are always there to morally support me in every exciting journey I take in life and my career.

To my family in Sweden: Thank you to my husband Gösta and to my four-legged best friend Lorentz, who, behind the scenes, provided patience and moral support, and reminded me to take breaks from this book project.

Finally, to all my followers and friends who show support in every adventure and journey I take in life, my career, and even my community engagements.

Thank you so much, Tack så mycket (Swedish), Maraming Salamat (Tagalog)

Foreword by Magnus Mårtensson

It is not an overstatement to say that *Learning Microsoft Azure* delivers one of the most important current IT skillsets in the market. This might be one of *the* most significant books you, on the way to the cloud, or as a learner already in the cloud, can read right now. Without cloud platforms, modern cloud-based services such as AI and large-scale batch computing, which are crucial for today's businesses, would not exist. There simply are not enough experts in the market that have the broad fundamental cloud platform skills and expertise needed to build these services. The book you hold in your hand is a comprehensive guide that covers the scope of what Azure, fundamentally, is about. These pages provide a solid leg up toward learning true cloud mastery, and the content provides strong competitive information for today's cloud-based IT landscape.

Writing this sort of book, covering so many different but related topics, and succeeding in boiling them down to a consumable size, is no small feat. Honored to write this foreword, I first thought—I could have written this book! As I read, I realized my perspective is not appropriate for authoring a terrific book about learning Azure. I have been an Azure expert and MVP since the first days of Azure, when the name was *Windows Azure*, and I know that it is quite challenging to explain to beginners what Azure is about. You must know what they need to learn and how they see the cloud in order to help them along. The person very much correctly placed to author this work is my friend and superstar Jonah. What she has accomplished with this book is right on the money. She clarifies a wide range of challenging topics in a way that someone starting their cloud journey will be able to consume. Even if you have experience with the cloud or are an expert, you'll still find value in this comprehensive resource due to its breadth and the resources and examples it provides. What Jonah has done is immensely difficult to pull off, *experto crede*, and she has delivered it with technical elegance.

Jonah is a shooting star in the tech space, with years of hard-won technical expertise, and her speaking engagements at conferences feel both strong and natural. As a leader in the Azure community space in Sweden she has

turned her natural curiosity and her work ethic into a set of skills she willingly shares with everyone who is eager to transition to the cloud or expand their migration. It's come as no surprise that Jonah has funneled her energy and passion into a book about Azure.

Logically, the book starts with covering the fundamentals, including what the cloud is and what the Azure cloud is. Through my experience, I have seen that nontechnical business leaders need to understand these foundational concepts in order to map their business needs to the benefits a cloud migration can offer. Beyond business leaders, everyone in a company focusing on cloud should read the first part of the book to lessen the gap between technical and nontechnical backgrounds. Strategically, all companies need to appreciate that the lofty goals of AI, ML, big data, and IoT, which are covered in **Part III**, stand on the foundations of Azure laid out in **Part II** of the book. Accordingly, focusing on your fundamental technical understanding is pivotal to building the future cloud services that will win business in a competitive marketplace. Using the cloud instead of traditional IT will quite likely lead to good results, but there is no guarantee that they will be great. Rather, greatness comes from how you use the cloud and what tools you employ to make your cloud usage truly effective. This includes practices like automation, autoscaling, and effective operations, to name a few. **Part IV** dives into the practices that take you from a good cloud to a great cloud, and **Part V** completes the cloud tour with the tools that are not only in the cloud but are needed to complement the cloud so you can take optimal and full advantage of the power of the cloud.

The name of the game in the cloud is knowledge. The more you know the better your Azure solutions will be! Red Adair said, "If you think it's expensive to hire a professional to do the job, wait until you hire an amateur." That is why this book, which offers a far-reaching overview of the professional cloud landscape, is so useful. It will help you to see a clearer picture and to pierce those cloud veils so that you can find the rainbow in the clouds and the gold at the end of it!

Think of this book as a fundamental handbook to the Azure cloud. You can never arrive at the end of the cloud, because clouds continue to change

shape forever. What you can do instead is use this book to gain an understanding of the nature of cloud computing. With this book you will become aware of all of the cloud's distinct characteristics and qualities. Guided by these pages you can take decisive steps straight toward a productive cloud. Congratulations on deciding to focus on learning Microsoft Azure, because you will be ahead of the game after reading this book.

Magnus Mårtensson

Loftysoft CEO, Azure MVP since the start of Azure, Microsoft Regional Director, expert consultant and advisor, global trainer, and keynote holder August 2023

Foreword by Thomas Maurer

Cloud computing is one of the most important and transformative technologies of our time. It enables organizations to innovate faster, scale more efficiently, and reduce costs while delivering better services and experiences to their customers and stakeholders. Microsoft Azure is a leading, open, flexible, enterprise-grade cloud platform that offers more than 200 products and services for building, deploying, and managing cloud applications and solutions. It supports a wide range of programming languages, frameworks, tools, and operating systems. It also integrates with other Microsoft products and services, such as Office 365, Power Platform, GitHub, and Visual Studio. Due to the breadth of its services, Microsoft Azure enables you to create anything from simple websites to complex enterprise applications, from mobile apps to artificial intelligence solutions, from data analytics to Internet of Things scenarios.

In this book, you will learn the fundamentals of cloud computing and Microsoft Azure from an experienced and passionate author who has been working with Azure for many years. Jonah Andersson is not only a software engineer and a DevOps leader but also a Microsoft Most Valuable Professional (MVP) and a community advocate who shares her knowledge and expertise with others through speaking, mentoring, and writing. She has been involved in several cloud migration and transformation projects, and she knows the challenges and opportunities that come with moving to the cloud. I have personally known Jonah for over a decade as a valued member in the Microsoft MVP program, industry leading Azure expert, and friend.

This book is designed to help you get started with Microsoft Azure and gain confidence in using its various services and features. You will learn how to choose the right cloud deployment model for your needs, how to migrate your existing applications or create new ones using Azure services, how to collaborate with your team using Azure DevOps, how to secure your cloud resources using Azure security services, and how to leverage the power of Azure serverless, IoT, and cognitive services to enhance your cloud solutions.

Whether you are a beginner or an intermediate user of Microsoft Azure, you will find this book useful and informative. It covers the essential topics and concepts that you need to know to succeed in your cloud journey. It also provides practical examples and exercises that you can follow along with to reinforce your learning. By reading this book, you will gain a solid foundation of cloud computing and Microsoft Azure that will help you advance your career and achieve your goals.

I highly recommend this book to anyone who wants to learn Microsoft Azure and cloud computing from a trusted and experienced author. Jonah Andersson has done a great job of writing a clear, concise, and engaging book that will guide you through the fundamentals of Microsoft Azure. I hope you enjoy reading this book as much as I did.

***Thomas Maurer,
Senior Program Manager and Chief Evangelist, Azure Hybrid Microsoft
August 2023***

Preface

Greetings from the Author

Hello, hej, and mabuhay to you, the reader of *Learning Microsoft Azure*!

As the author, I want to thank you for picking up this book and choosing to read it.

I appreciate that you invested time, money, and effort to read this book to help you learn about Microsoft Azure. Our time is precious and investing time in learning is golden. I hope you find this book a valuable resource for your learning journey. Application development, DevOps, and engineering with cloud technologies have been the most exciting experiences in this modern era of artificial intelligence. We, as innovators and builders of these technologies play a great role of the future of our modernization and digitalization.

We have gone from the age of huge steel mainframe computers to massive virtual modern resources and technologies on the cloud. I remember when I was studying computer science, I had to use several of those low-memory 3½-inch floppy disks to save a programming project on the computer. Today, we don't have to worry about that. As much as the industrial revolution forever changed the manufacturing industry and our access to consumer goods, digitization through cloud computing has transformed how we live and work.

Microsoft Azure is a cloud computing platform close to my heart. For many years, I have been developing applications hosted on premises and on the cloud, working with the modern DevOps practices we have today. Working several years in the IT industry in different roles, I have learned a lot and gained significant hands-on experience, both technical and soft skills, and I am inspired to share knowledge about Azure.

Azure is a global cloud provider serving millions of organizations, customers, and users who are building modern applications. This cloud platform brings powerful benefits, not just to the business, but to diverse members of the organization, from leadership to project managers, clients, and engineering teams.

Developing and maintaining enterprise systems is my daily work routine. The experience I gained in the software engineering and working in the IT consulting industry in different areas has helped broaden my technical knowledge and toolbox, which I am now able to share with you.

The Cloud Migration Journey to Azure That Leads Me to You

I was once involved in a cloud migration project to Azure to help develop, migrate and rehost an old legacy .NET application hosted on on-premises servers. That legacy application was becoming outdated; it was also not documented and had different code patterns and styles developed by different types of developers. Alongside a small team assisting me, I had full responsibility for moving it to a cloud computing platform. Although it is not really Agile to do it alone without a dedicated team, it was part of my job, and I also wanted to take the opportunity of learning from new challenges. So, I could say that I was like a superwoman on a cloud migration mission.

From designing the system architecture to restructuring the databases, fixing the technical debts, fixing data quality issues, programming, and even creating documentation, I put my heart into it. But that cloud migration project was a fiasco.

After trying different migration alternatives (lift and shift, refactoring code, and re-architecting the infrastructure), we determined that the system needed to be rebuilt as though new. Unfortunately, that project was stopped just when we were close to success. It was not deployed to production because the organization lacked a cloud migration strategy, awareness of

the significant benefits of using cloud technologies, and knowledge about the cloud, among many other organizational factors.

It was a project that didn't make it to the finish line. Still, it was one of the most significant experiences in my cloud engineering career. I learned a lot from that migration to cloud project, including the mistakes we made in that project and other lessons to take forward.

Just like any other project, and in life, there are no regrets, just lessons learned.

It is my hope that you as the readers of this book will not commit the same mistakes we did in that cloud migration project. This is one of the reasons why this book exists. It is my hope that learning Microsoft Azure will help you prepare and take smart steps in any cloud migration project you will be involved in.

My cloud migration journey as a developer, and partly the role of being a cloud architect, is one of the reasons I am passionate about sharing knowledge of cloud computing and Azure, especially to those still starting their journey. I inspire knowledge through public speaking at tech conferences, meetups, etc. I also apply what I teach in the career roles I do every day.

It is an honor to spend part of my late nights, after-work hours, and weekends writing this book for you. I hope that the learning you will explore and gain from this book will come in handy in your work, career development, cloud migration journey, and contribution to your organization's cloud engineering projects.

One day the information you learn about Azure might be outdated, but you are on the right track starting a great journey with me through this book.

The entire book has a lot of learning and inspiration in store for you. It is my hope that after you read this book, you will want to learn more by doing and applying it to your existing and future projects!

Why I Wrote This Book

My cloud migration journey as a developer is the primary reason I wrote this book. I also want to inspire and help others to truly plan and get ready to move their workloads and applications to the cloud. Although the clouds are really exciting and beautiful, we cannot just move up there! We need to prepare because by failing to prepare we are creating risks to fail.

When I was studying computer science in the 1990s, my first thesis was about how the invention of internet technology helped local communities and our society. I was curious to finish my thesis and truly understand what the internet was really about and how people adapted to its early phase. The result of my thesis research from community surveys I conducted inspired me to study more about it, which led me to my journey of becoming a tech enthusiast and programmer. Since then, I have become fascinated with how new technologies affect us in our daily routines and work.

As a developer (focused mainly on Microsoft technologies) and cloud platform and DevOps engineer I create, build, and develop technical solutions with modern technologies, including the cloud services offered by Azure. I collaborate in application development, cloud infrastructure management, and automation with DevSecOps.

The more I collaborate with different teams in different areas and the more problems I solve, the more technical skills and knowledge I gain to keep me up-to-date with evolving modern technologies.

I am passionate about sharing this knowledge with others, especially those who want to gain fundamental learning of cloud computing and Microsoft Azure.

This book will help IT professionals, project teams, software developers, and cloud engineers choose the appropriate cloud service to use in Microsoft Azure. This knowledge will help solve your organization's customer use cases and business requirements.

Who Should Read This Book

This book is an essential learning reference book for anybody who wants to learn about the vital cloud concepts and cloud computing services provided by Microsoft Azure, regardless if you are a beginner or at an intermediate level in this field.

The content of this book will also help you gain the technical knowledge to plan, design, and develop applications and modern technological solutions and migrate existing workloads and systems to the cloud using Azure.

This book is for anybody with a technical background with career roles like software developers, cloud engineers, or cloud solution architects.

Management or leadership roles in an IT organization, such as IT project managers, technical sales managers, and scrum masters, would also benefit from learning the concepts of cloud computing in Azure. Also, teams working with traditional and on-premises legacy applications or systems will gain essential insight into designing and developing solutions in the cloud platform through the Microsoft Azure service.

This book is also ideal for IT professionals, software developers, and aspiring cloud engineers who want to earn Microsoft Certifications for Azure, such as **AZ-900, Microsoft Azure Fundamentals**; **AZ-204 Developing Solutions for Microsoft Azure**, **AZ-400, Designing and Implementing Microsoft DevOps Solutions**; and **AZ-305, Designing Microsoft Azure Infrastructure Solutions**. Please note that the exam names of the Microsoft Azure certifications may change at any time.

This book will give you the concepts you need to get started with cloud-native development, administration, data engineering, DevOps, analytics, migration strategies, and other exciting solutions in Azure.

What You Will Learn

By the end of this book, you will understand the following:

- Essential and fundamentals concepts about cloud computing and its benefits to you as a developer and/or IT professional and to your organization
- The core concepts and fundamentals of Microsoft Azure as a public cloud provider to help you build modern cloud solutions and applications
- The different Microsoft Azure technologies that will help you and your organization develop, transform, modernize, and migrate to modern cloud environments
- A comprehensive overview of the different cloud technologies in Microsoft Azure that will help you choose the right cloud service for your demands, use cases, software development, and cloud development projects
- A jump-start guide on how you can start developing cloud solutions and accelerate your career as an Azure developer or DevOps, cloud, or platform engineer
- Start developing modern cloud services, applications, and solutions in Microsoft Azure environments using your desired and supported programming languages, frameworks, and tools
- Learn about the different cloud security tools, including identity and access management (IAM), in Azure to secure your cloud resources, applications, and users
- Learn how you can integrate cloud technologies with other services, APIs, and third-party services
- Gain knowledge about services in Azure that will help you implement cloud security and DevSecOps practices in your organization
- Get practical options and learn from best practices on the important things you need to consider when you migrate existing legacy

applications to a cloud platform like Microsoft Azure using different migration tools and services

- Get information about the recent tools and cloud adoption frameworks in Azure that will help you in multi-cloud or hybrid cloud environments
- Develop with Azure by learning the best practices, different programming languages, and modern developer tools like Azure Developer CLI (azd), GitHub CLI, GitHub Copilot, GitHub Codespaces, and Microsoft Dev Box—all of which you can use to build cloud-native applications and solutions such as infrastructure as code (IaC), serverless, containers, and other cloud technologies in Azure.

NOTE

Learning Microsoft Azure is your guide as you work with Microsoft Azure. I believe in the **idea of learning by doing**.

Check out a blog post I wrote about this topic: “**Continuous Learning—An Integral Part of A Programmer’s Development**”.

Invest time in learning the fundamental concepts and doing some hands-on work.

Navigating This Book

Part I: Cloud Computing and Microsoft Azure Fundamentals

In this book's introductory part, you will learn about cloud computing and Microsoft Azure fundamentals.

Chapter 1, “Cloud Computing Fundamentals”, provides an introduction to the essential concepts of cloud computing, how it works, the different types of deployment models, understanding the types of cloud, what CapEx and OpEx are in cloud computing, and the benefits of utilizing cloud computing in businesses, IT organizations, society, and software engineering.

Chapter 2, “Microsoft Azure Fundamentals”, focuses more on the theoretical and technical concepts of Microsoft Azure as a public cloud platform. Learn about the Microsoft Azure core components and the different cloud services categorized by their purpose.

By the end of **Part I**, you will gain vital knowledge and foundational concepts of cloud computing and Microsoft Azure.

Part II: Compute, Networking, Storage, and Databases

This second part of the book and its chapters focus on the different technologies in Microsoft Azure, grouped into categories.

Chapter 3, “Microsoft Azure Cloud Compute Services”, explores some of the Microsoft Azure compute services such as Azure Virtual Machines; container services like Azure Container Instances, Azure Container Registry, and Azure Container Apps; Azure App Services for web and mobile applications; serverless cloud solutions with Azure Functions; Azure Static Web Apps; and more.

Chapter 4, “Microsoft Azure Cloud Networking”, covers cloud networking and services in Microsoft Azure, including Azure VNet, DNS, Azure Firewall, Azure Front Door, ExpressRoute, Virtual Network, VPN Gateway, Application Gateway, Load Balancer, Internet Analyzer, and more. This chapter briefly mentions Azure Orbital, a fully managed ground station as a service (GSaaS) solution in Azure.

Chapter 5, “Microsoft Azure Cloud Storage and Databases”, provides a technical overview of the different cloud storage options and databases (both SQL and NoSQL) in Microsoft Azure. You will learn about cloud storage concepts and services, create databases, and find applicable best practices for Azure SQL Databases, Azure Cosmos DB, Database for MySQL, Azure SQL Servers, Redis Cache in Azure, Azure Storage, Data Share, and Manage Disks.

By the end of **Part II**, you will be ready to start developing and building applications for the cloud using the compute, data storage, database, and networking options in Azure.

Part III: Artificial Intelligence (AI), Machine Learning (ML), Big Data, IoT, and Security

Chapter 6, “Artificial Intelligence, Machine Learning, and Cognitive Services in Azure”, focuses on the valuable concepts you need to know about artificial intelligence (AI), machine learning, Azure OpenAI Service, and Cognitive Services in Microsoft Azure. This chapter also covers the importance of responsible and ethical AI.

Chapter 7, “Big Data, Reporting, and Analytics Services in Azure”, explores the Big Data, Reporting, and Analytics Services in Microsoft Azure. It will include what you need to know about data analytics, big data, analytics, and reporting services using Power BI, Azure Stream Analytics, Data Lake Analytics, Azure HD Insights, and Azure Analysis Services. This chapter will help you learn the critical concepts for working with extensive complex data using Azure services and big data tools.

Chapter 8, “Cloud IoT and Maps Services”, covers the Azure solutions for IoT (Internet of Things), Maps Services, and Cognitive Services in Azure. You will learn about Azure IoT Hub, IoT Edge, Azure Maps, Azure Spheres, and Remote Rendering services. You will learn how to use some of the Azure IoT kits available for developers such as the Azure Developer IoT Dev Kit to start IoT development with Microsoft Azure.

Chapter 9, “Azure Security, Identity Management, and DevSecOps”, dives into identity and access management (IAM), compliance, cloud security, and DevSecOps in Microsoft Azure, which is important in protecting and securing your applications and cloud workloads. You will learn about Microsoft Entra ID. Azure security services like Azure Key Vault, Azure Sentinel, Azure Firewall, Microsoft Defender for Cloud will be explored, along with other Azure security monitoring features for networking, data, and compute.

By the end of **Part III**, you will have gained technical knowledge and hands-on experience with the different technologies for developing applications with AI, ML, big data, and cloud security in Microsoft Azure. These will

help you choose what Azure technology to use for your business requirements and how to develop solutions with them.

Part IV: Integration, Infrastructure, and DevSecOps

This book's fourth part will teach you how to integrate Azure technologies with other services, use infrastructure as code solutions, and set up your application using Azure's deployment technologies.

Chapter 10, “Azure Cloud Integration Services and Tools”, focuses on Microsoft Azure integration services. In this chapter, you will learn about the different cloud technologies you can integrate with your cloud-native applications, including existing ones (on the cloud or on premises). This chapter covers Azure API Management, Azure Logic Apps, Azure Web PubSub, Azure Service Bus, Azure Event Grid, Azure Logic Apps, etc.

Chapter 11, “Cloud Infrastructure, DevOps, and Monitoring in Azure”, guides you on how to develop systems or applications with modern DevOps tools. You will learn what DevOps means, and about Azure DevOps, a comprehensive suite for team collaboration for developers and IT operations. You will also learn about automating development processes using CI/CD and source code version control, what Azure Pipelines, GitHub, and Azure DevOps are, and how Azure Monitor and Application Insights are used for troubleshooting Azure resources. You will also learn about other cloud technology services like Azure DevTestLabs, Azure Bicep, ARM templates, and more. You will also learn about infrastructure as code (IaC), policy as code, and configuration as code and how they help with the automation of deployments.

After reading the chapters in **Part IV**, you will have knowledge and development skills to help you work effectively as a cloud developer using cloud integration and automation options. You and your organization's Agile team will also be able to collaborate and work effectively using the great features of Azure DevOps, DevTest Labs, GitHub, etc.

Part V: Governance, Migration, Architecture, and Development Tools

The fifth section of this book will give you essential knowledge and tools for adopting and migrating successfully to the cloud.

Chapter 12, “Cloud Management and Governance in Azure”, walks you through the essential concepts regarding cloud management and cloud governance in Microsoft Azure. For example, it covers automation in the cloud, Azure Advisor, backup using Azure Backup, Azure Blueprints, Azure Policy, Azure Monitor, and other known solutions for hybrid and multi-cloud.

Chapter 13, “Cloud Migration, Hybrid, and Multi-Cloud Solutions in Azure”, discusses cloud migration, cloud transformation, and architectural concepts in Microsoft Azure. You will learn the vital facts needed for adopting, transforming, or migrating to Azure. Find out the best practices and a helpful list of tools you can use when moving on premises and handling legacy applications. This chapter also highlights the importance of having Microsoft Azure’s Well-Architected Framework.

Chapter 14, “Cloud Development Tools for Azure”, is more focused on cloud development with Azure. This chapter discusses the different language and cloud development tools you can use to get started with developing modern and resilient applications and solutions on Azure. It covers IDE tools such as Visual Studio, Visual Studio Code, Azure Cloud Shell, GitHub Copilot, Codespaces, GitHub CLI, Azure Developer CLI (azd), Microsoft Dev Box, and many more. You will learn the different technologies that will prepare you for your projects and career development.

After reading the chapters in **Part V**, you will have an essential understanding of Microsoft Azure cloud governance and management. These are important in designing and developing cloud solutions. You will also capture a broad understanding of the Microsoft Cloud Adoption Framework (CAF), Well-Architected Framework for Azure (WAF), Azure

Migrate, and Microsoft Assessment tools that can assist your cloud migration projects. You will also have learned from lessons I learned from a cloud migration project experience in migrating .NET legacy on-premises applications to Azure. You will deepen your knowledge on cloud development using the development and DevOps tools well integrated with Microsoft technologies and Azure. You will also learn some of the different ways to migrate and develop your existing IT infrastructure and on-premises applications as hybrid or multicloud on Azure.

Check Your Knowledge

At the end of each chapter, I will provide a short *Check Your Knowledge* section with a maximum of five learning check questions related to the topics discussed. These questions will be a useful review challenge for you to confirm what you have learned.

Learn by Doing (Try It!)

As a developer, I love the concept of learning by doing. I believe in theoretical learning enhanced by hands-on practices. Therefore, each chapter in this book also ends with a section called *Learn by Doing (Try It!)* where I provided a recommended list of quickstart guides and how-to links to further your hands-on learning related to the topic discussed in the chapter.

Learning Resources and Further Readings

Learning Microsoft Azure aims to provide the most important foundational knowledge you need to know. Considering that we have different levels of experience and knowledge, you probably want to explore the discussed topic in more depth. Therefore, at the end of each chapter, I will provide a short list of recommended learning and further reading resources related to the topics discussed. Most references are linked to the most recent Microsoft documentation for the specific technology or resource mentioned in this book.

What This Book Is Not

This book is not an advanced level book for each cloud technology service in Microsoft Azure. Hands-on examples are provided but are not intended for those who already work with Azure at highly advanced levels.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.

TIP

This element signifies a tip or suggestion.

NOTE

This element signifies a general note about the topic discussed.

WARNING

This element indicates a warning or caution.

Using Code Examples

Supplemental material is available at <https://github.com/learningazurebook>. For more information about the book, updated code examples, and blogs from the author, please visit the book's website: <https://learningmicrosoftazure.com>.

If you have a technical question or a problem using the code examples, please email bookquestions@oreilly.com.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Learning Microsoft Azure* by Jonah Carrio Andersson (O'Reilly). Copyright 2024 Jonah Carrio Andersson, 978-1-098-11332-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Online Learning

NOTE

For more than 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit <http://oreilly.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-889-8969 (in the United States or Canada)

707-829-7019 (international or local)

707-829-0104 (fax)

support@oreilly.com

<https://www.oreilly.com/about/contact.html>

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *<https://oreil.ly/learning-microsoft-azure>*.

For news and information about our books and courses, visit *<https://oreilly.com>*.

Find us on LinkedIn: *<https://linkedin.com/company/oreilly-media>*

Follow us on Twitter: *<https://twitter.com/oreillymedia>*

Watch us on YouTube: *<https://youtube.com/oreillymedia>*

Acknowledgments

I want to acknowledge and thank those who helped, supported, and contributed to making this book a success from start to finish. This book would not have been possible without my family, friends, and the incredible people in the tech and cloud communities who helped and contributed.

Special thanks to my friends in the community who read the online early release version of this book and left some good feedback: Billy Hollis, Lior Yantovski, Peter Lee, Richard Vaughan, Keith Atherton, and many others. Also thanks to the technical reviewers who helped the quality of my book. I appreciate the time invested by Peter De Tender, Håkan Silfvernagel, Micheal John Peña, Nikos Delis, Stefano Demiliani, Peter Lee, Sagar Rastogi, Vladimir Serykh, Szabó Attila, Ian Santillan, George Grammatikos, Dipal Choksi, Alexander Wachtel, Dave McCollough, George Mount, Gerald Versluis, Demiliani Stefano, Robin Smorenburg, and other reviewers that I may have missed mentioning.

To Andrew Urwin and Freek Berson who contributed their input about working with DevOps and infrastructure as code (IaC) with the Bicep Language in Azure in **Chapter 11**.

Thanks to experts and friends in the community such as John Kilmister (also a technical reviewer), Alexey Polkovnikov, Ryan O’Connell, Håkan Silfvernagel, Adrienne Braganza Tacke, Goran Vuksic, Andrew Urwin, Ezhilarasi Chezhiyan, George Mount, Tiago Costa, Marilag Svennevig, Sasha Kranjac, Hamida Rebai, and Kristina Devochko who provided their quotes at the beginning of each chapter of this book.

Thanks to Jill Leonard, Jennifer Pollock, Megan Laddusaw, Elizabeth Kelly, Suzanne Huston, Helen Codling, Rita Fernando, Beth Richards, Aleeya Rahman, and the rest of the O’Reilly Media team who supported me and helped me when I had questions during the development of this book.

A *big* thanks to the inspiring forewords of Magnus Mårtensson and Thomas Maurer, and afterword of Maxim Salnikov, who also helped in the technical review of this book.

Finally, I would like to thank Karen Montgomery for designing amazing covers for O’Reilly books, including the amazing Hyacinth Macaw cover of this book. I fell in love at first sight with the colors and character of this amazing, gentle giant parrot of the world. If you are curious about the cover, make sure you read the Colophon section at the end of this book.

Special mention to anybody who helped me through the journey of this book that I probably missed and most of all to those who followed and supported my writing journey. Thanks to all of you!

Part I. Cloud Computing and Microsoft Azure Fundamentals

In this introductory part of the book, you will learn about the fundamentals of cloud computing and Microsoft Azure. This part provides an introduction to the important concepts of cloud computing, how it works, and the different types of deployment models. It gives you a general perspective to understand the types of cloud computing. This part also focuses on the theoretical and technical concepts of Microsoft Azure as a public cloud platform of Microsoft.

Chapter 1. Cloud Computing Fundamentals

Over the last decade, cloud computing has changed the way we build and deploy software. It is now easier than ever to deploy highly scalable, resilient, and secure solutions to a global audience, in many cases at a fraction of the previous cost.

—John Kilmister, Software Architect and Microsoft Azure MVP

What Is Cloud Computing?

Before delving into learning Microsoft Azure, it's essential first to understand cloud computing. Cloud computing is a significant technological innovation that delivers various services through the internet, including web servers, databases, data storage, virtual machines, applications, network infrastructure, security tools, software, and other IT infrastructure. *Cloud computing* refers to the virtual storage and access of data and information over the internet, with the actual computing processing transpiring in the cloud.

Understanding the foundational principles of cloud computing is essential for designing and developing solutions in the cloud. As a public cloud platform, Azure provides secure, scalable, reliable, cost-effective, and easy-to-manage ways to build on the cloud. By leveraging Azure, organizations and users developing with it can take advantage of various cloud services and tools to innovate and solve complex business problems.

As a modern, game-changing technology, cloud computing enables businesses, organizations, and teams to access a wide range of IT resources on demand through the internet. With Azure, we can build and deploy cloud-based solutions that are secure, scalable, and cost-effective.

JOHN MCCARTHY'S INTERESTING CLOUD COMPUTING PREDICTION

John McCarthy, an American computer scientist and cognitive scientist known as the **father of artificial intelligence**, gave some interesting insights about cloud computing in his speech at MIT's centennial celebration in 1961. He suggested that computing could be sold like a public utility, just like water or electricity.

Cloud computing makes life easier for us. A practical example is the possibility of saving our photos, videos, and files into cloud storage with virtually unlimited capacity instead of keeping them on a local storage device with limited storage. Another benefit is virtualizing web servers and databases instead of having the physical infrastructure or servers in costly data centers.

Software engineering and modern IT innovation are a few significant advantages of the cloud. Adopting and implementing cloud computing provides a fully managed cloud computing infrastructure and services with the benefits of scalability, autoscaling, availability, and performance at a flexible global scale.

The different cloud computing solutions give us tools and capabilities to handle peak loads based on demands at any time on a global scale. This is complicated, expensive, and time-consuming in an on-premises environment. Azure seamlessly handles scenarios like this through **horizontal or vertical scaling** or a combination of both.

Benefits of the Cloud in Software Engineering and IT

Cloud computing and software engineering are both evolving rapidly. The evolution of computing in the cloud, along with innovations like machine learning (ML), the Internet of Things (IoT), edge computing, quantum computing, and big data, has caused an increased demand for skillsets and people who can work in these technologies and platforms in the cloud.

Computing in the cloud enables software developers, engineers, and even IT professionals to create, build, test, and deploy technical cloud solutions productively, effectively, and securely. Software engineering teams still need to be expanded to work and develop on premises, and they encounter technological gaps. They may experience the risks of missing the advantages of developing cloud computing systems.

Engineering teams, business teams, and organizations may risk missing the great features, benefits, and capabilities that cloud computing provides. Cloud engineering offers better speed of development, testing, maintainability, automation, scalability, and so much more.

With the advancement of modern automation processes like **infrastructure as code (IaC)** becoming available for cloud infrastructures, benefits such as infrastructure automation are making software development easier for developers and DevOps teams. Through the IaC approach, there will be more consistency and routines for configuring systems with the capability of replicating systems to several environments.¹

In addition to this, solutions and **platforms using low-code/no-code** like **Power Apps, Azure Logic Apps, AI Builder**, etc., on the cloud also help IT professionals with fewer programming skills and less experience to build modern and smart applications quickly on demand. Infrastructure as code technologies and low-code/no-code solutions will be discussed later in this book.

To complement the benefits the cloud has for engineering teams, IT managers can easily manage their projects and collaborate with their teams by working Agile with available modern and remote collaboration tools in the cloud. For example, Azure DevOps is an all-in-one collaboration suite used by operations and development teams to plan, build, test, deploy, and monitor applications. It helps cover the entire application lifecycle, including Agile project planning, source code versioning, continuous integration and continuous delivery (CI/CD), testing plans, artifacts, and integrations. We will dive more into Azure DevOps, cloud development

integration tools, and infrastructure automation and management in **Chapter 11**.

Digitalization and modernization come with great benefits; however, they also come with challenges. These include the challenges of preparing, transforming, and adapting to fast-changing and evolving technologies. These barriers can be handled by learning the foundations of cloud computing. Instead of having our resources like databases, applications, servers, or infrastructure in physical data centers or on premises, we have these resources on the cloud or the internet.

NOTE

In cloud computing, a data center is a physical facility that houses many servers and other computing equipment. Consider a data center a place or physical location meant to provide a centralized place for storing and managing servers for databases and applications that users can access remotely over the internet.

Data centers are the backbone of cloud computing. For example, in Azure, they are the infrastructure necessary to deliver cloud-based services, including storage, computing power, and network connectivity. A data center's physical layout and design are planned to ensure high availability, security, and energy efficiency. They are equipped with redundant power and cooling systems, backup generators, and other measures to ensure continuous operation, even during a power outage, bad weather conditions, or any type of disruption. They also employ strict security measures with biometric authentication of whoever enters the premises. Security systems detect intrusion to protect the resources and assets within the data center premises.

Businesses and organizations consider cloud computing technology a good and strategic option because of its speed, reliability, financial savings, productivity, efficiency, security, performance, and more. By the end of this chapter, you will learn the specific benefits cloud computing has for different categories.

Cloud computing has been a popular option for many because of the growing demand and evolution of technological innovations over the past decades. Organizations want to improve and modernize their systems to innovate with the new trends in technology. Computing in the cloud is

expanding and continues to grow; our mission to increase sustainability is also one of the great drivers of cloud innovation.²

Cloud Computing Versus Virtualization

Cloud computing and virtualization both create useful virtual environments. Hosting compute and data resources on the cloud is better than a virtual machine for a web server or databases hosted in an on-premises environment. Hosting in the cloud provides efficiency, flexibility, reliability, global scalability, and security.

The cloud is an environment, while virtualization is a technology that enables us to virtualize hardware to create and simulate several machines or dedicated resources. The cloud is an IT environment that pools and shares scalable resources across a network. Cloud environments are created to activate the extraordinary capabilities of cloud computing, such as running workloads within it.

NOTE

Cloud computing provides on-demand resources for computing, storage, network, platform, web applications, and infrastructure over the internet or cloud. These are pools of virtual services and resources hosted on the cloud, which is accessible by its users anywhere in the world. Cloud administrators can manage these cloud resources through self-service administrative portals—for example, Microsoft Azure Portal for Microsoft cloud resources.

Cloud Hypervisor: The Key to Virtualization in the Cloud

Hypervisor technology is emerging as a vital tool in virtualizing resources and is driving modern innovation in cloud environments. Hypervisors make resources and applications in the cloud available to their users remotely. Accessing and managing cloud resources over the internet gives organizations and IT better control in managing their systems, applications, data, and infrastructure in the cloud environment. Hypervisor technology is illustrated simply in [Figure 1-1](#).

The emerging transformation to digitalization and rising demand for better service expectations are resulting in building more modern and reliable applications. For these reasons, organizations are considering migrating or are already migrating their enterprise applications from on-premises virtual machines to cloud environments.

An excellent example of the use of hypervisors in cloud computing is the Microsoft Azure cloud platform. Microsoft uses a native hypervisor on Azure called Azure Hypervisor. It enables deployments of virtualized machines, web servers, database servers, enterprise applications, web services, etc., on the cloud. The hypervisor was initially based on Windows Hyper-V. It provides various virtualization deployment, management, monitoring, and **security features**.

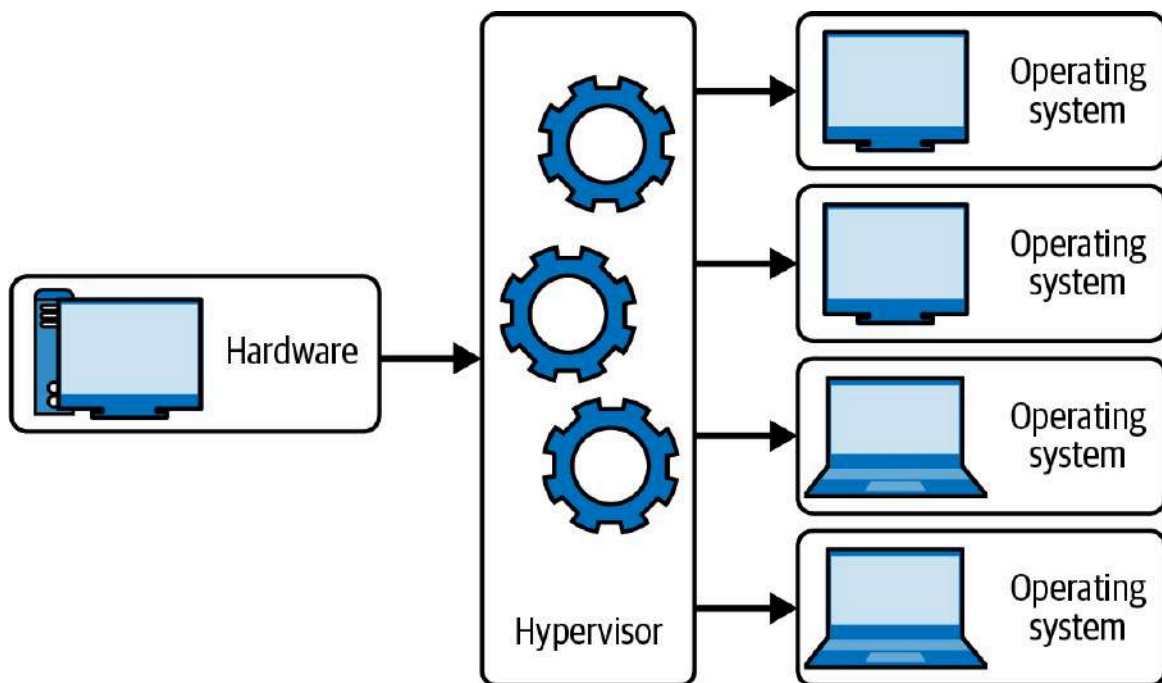


Figure 1-1. Hypervisor technology

Even with all of the benefits just described, migrating existing enterprise applications to the cloud is a challenging journey. It requires careful planning, implementation of good strategy, and more resources to rearchitect or rewrite systems or applications for cloud upgrades. Through the technology of virtualization with the hypervisor, it is possible to migrate

existing on-premise workloads to any cloud computing platform faster while investing less time, money, and resources.

NOTE

Virtualization technology simulates resources and environments from a single physical hardware system. Behind this technology is the software called a **hypervisor**, which has the capability to distribute a system into secure and distinct environments known as virtual machines (VMs). The VMs we use on our traditional web servers rely on the ability of a hypervisor. VMs are emulations of computers running on top of a hypervisor.

Today, containerization is one of the better alternatives to VMs. Unlike a VM, a container is a lightweight, portable, and isolated software unit that enables us to run multiple containers on a single host machine. Containerization allows us to run applications and services on the cloud.

In the later chapters of this book, you will learn more about Azure's different virtualization, infrastructure, and container solutions. In **Chapter 13**, you will learn more about Microsoft Azure cloud migration concepts and solutions.

Evolution of Cloud Computing

Earlier computing technologies were mainframe computers, which provided extensive computational capabilities. Mainframes were powerful, highly reliable, and specialized for large data movements and massive input/output (I/O) operations. Large organizations mostly used them for bulk data processing. Mainframes worked on batch processing.

There were different stages of earlier computing before we started using dynamic cloud platforms like Microsoft Azure. However, the early mainframe computer systems are similar to modern cloud computing platforms.

For example, both use a client-server model and thin clients. The reason why many organizations and institutions are migrating to the cloud is

because of cost savings, increased productivity of IT teams, speed, availability, flexible scaling, efficiency, performance, and security. The earlier cloud computing technologies have evolved to create more dynamic technology solutions and offerings like the public cloud, which platforms like Azure provide.

Mainframe computing

Mainframe computing utilizes large, high-performance computers to complete critical tasks. It is also referred to as **big iron** and uses a single unit of hardware like a huge mainframe box with several processors, centralized storage, and a large amount of memory. IBM was the pioneer of mainframe computers, which are mainly client/server-based systems. They offer high performance and significant processing power to handle massive data like transactions and calculations in real time.

Mainframe characteristics include utilizing time sharing, high security, and support for **batch processing**. The drawbacks of the mainframe are that they are expensive to maintain and do not support the X86 architecture. A significant challenge of mainframe computers is the limited amount of skilled engineers who can maintain them.

Cluster computing

Cluster computing consists of tightly coupled computers (also known as nodes) that work together to reach a single goal and purpose: to execute tasks. A cluster's components are connected through a closed group of local area networks (LANs). When multiple computers are clustered, they share the computation tasks like a distributed system.

Cluster computing is commonly used for implementations of business requirements and optimization of performance: types include high-availability, high-performance, and load-balancing clusters.

Cost-effectiveness, scalability, high availability, and speed processing are some of the benefits of using cluster computing. These can be implemented in real-life use cases like search engines, earthquake simulation, and weather systems.

Earthquake simulation is an exciting and valuable application because earthquake dynamics are significant and challenging in geophysics and computer modeling because of their highly nonlinear nature. To learn more about these earthquake simulation studies utilizing cluster computing, please read "*NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids*" and "Study Uses Supercomputers to Advance Dynamic Earthquake Rupture Models".

As noted, cluster computing is composed of multiple computer systems called nodes. These nodes are used together to execute tasks. This type of computing has expanded considerably in modern days.

Azure offers **high-performance computing (HPC)**, which integrates resources from storage, computing, and networking with workload orchestration.

Grid computing

Grid computing is a subset of parallel and **distributed computing** in which clusters of computers and other loosely coupled computers perform a large task. The computer resources can be geographically spread out in different locations or in several computing clusters that form the grid. The advantage of this is that data is processed quickly because the data is stored on all computers in the data grid.

The computer systems on the grid in the same network work together and serve as a virtual supercomputer. All systems in the grid use compute resources like storage capacity and processing power.

Cloud computing

Cloud computing is considered **fifth-generation computing**. The evolution of mainframe computing, grid computing, and cluster computing created an enormous path of accelerated innovation that drove and enabled the modern computing we have today. The technology of cloud computing is widely used, and we are continually exploring more of its capabilities in our modern digitalization.

Based on Foundry's *Cloud Computing Study 2022*, over 40% of companies are planning to migrate their data integration, disaster recovery, business intelligence (BI), data warehousing, data analytics, and backup to the cloud.

Our Journey to the Modern Cloud

Advances in digitalization are amongst the most impressive and impactful technological innovations in our history. When I was studying computer science in the 1990s, I used small-capacity storage floppy disks to save my documents and photos.

Using the cloud, we have many alternatives and an enormous capacity to collect and store data. The data we store on the cloud is portable and accessible anywhere. **Portability of accessing data** and getting the information we need when we need it, wherever we are, provides huge benefits and is also practical.

Modern cloud computing delivers reliability, scalability, agility, cost savings, and portability to our applications and resources globally. *Azure compute services*, which we will explore in **Chapter 3**, enable us to build, manage, and scale cloud computing applications and services.

Cloud Computing Deployment Models

Cloud computing deployment models give us a descriptive overview of cloud computing platforms and their various categories, helping us identify essential facts like who has access to the cloud, how it is hosted, and what is implemented.

Figure 1-2 shows some commonly used cloud deployment models: public cloud, private cloud, and hybrid cloud. Other deployment models, such as community cloud and multi-clouds, are trending and in demand. These deployment models work the same way by virtualizing servers' computing power into segmented applications with speed, reliability, scalability, and massive storage capacities.

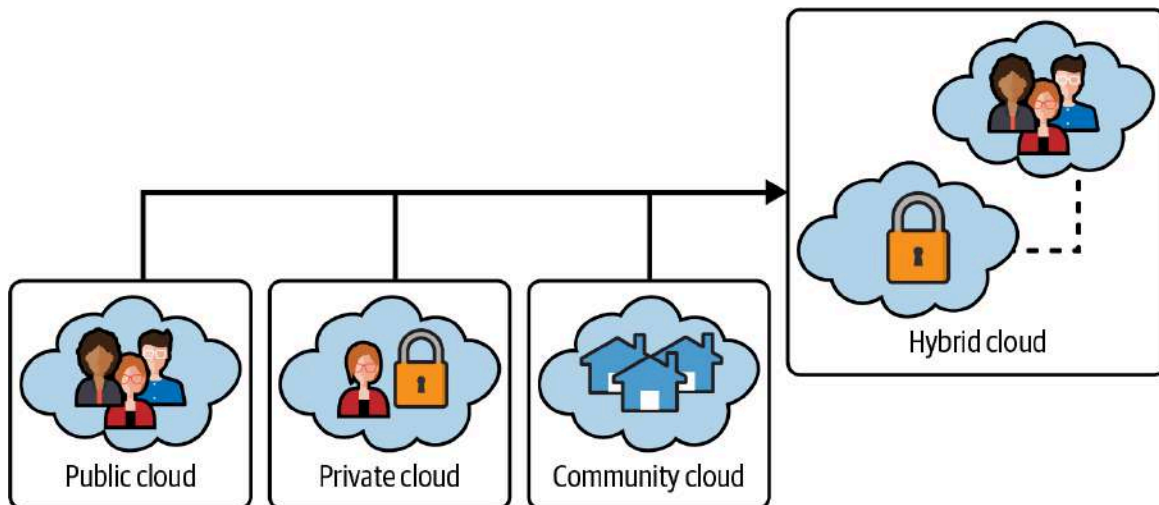


Figure 1-2. Different types of cloud computing models

Public Cloud

A public cloud infrastructure is available to the public or any organization using or selling cloud services. A public cloud platform is a service provided by cloud providers like Azure. The public cloud vendors offer cloud storage and computing resources (operating systems, CPU, memory, storage, web servers, applications, or databases) that are securely shared among its customers with other organizations or tenants of the cloud. The public cloud is offered to its users for a subscription fee or on a pay-as-you-go basis.

For example, in Azure, you can start using that platform as an individual by signing up for a free account with a **pay-as-you-go** type of subscription with included free Azure services. Many organizations globally have adapted and evolved to use the public cloud as their central platform for IT infrastructure and services, as illustrated in **Figure 1-3**.

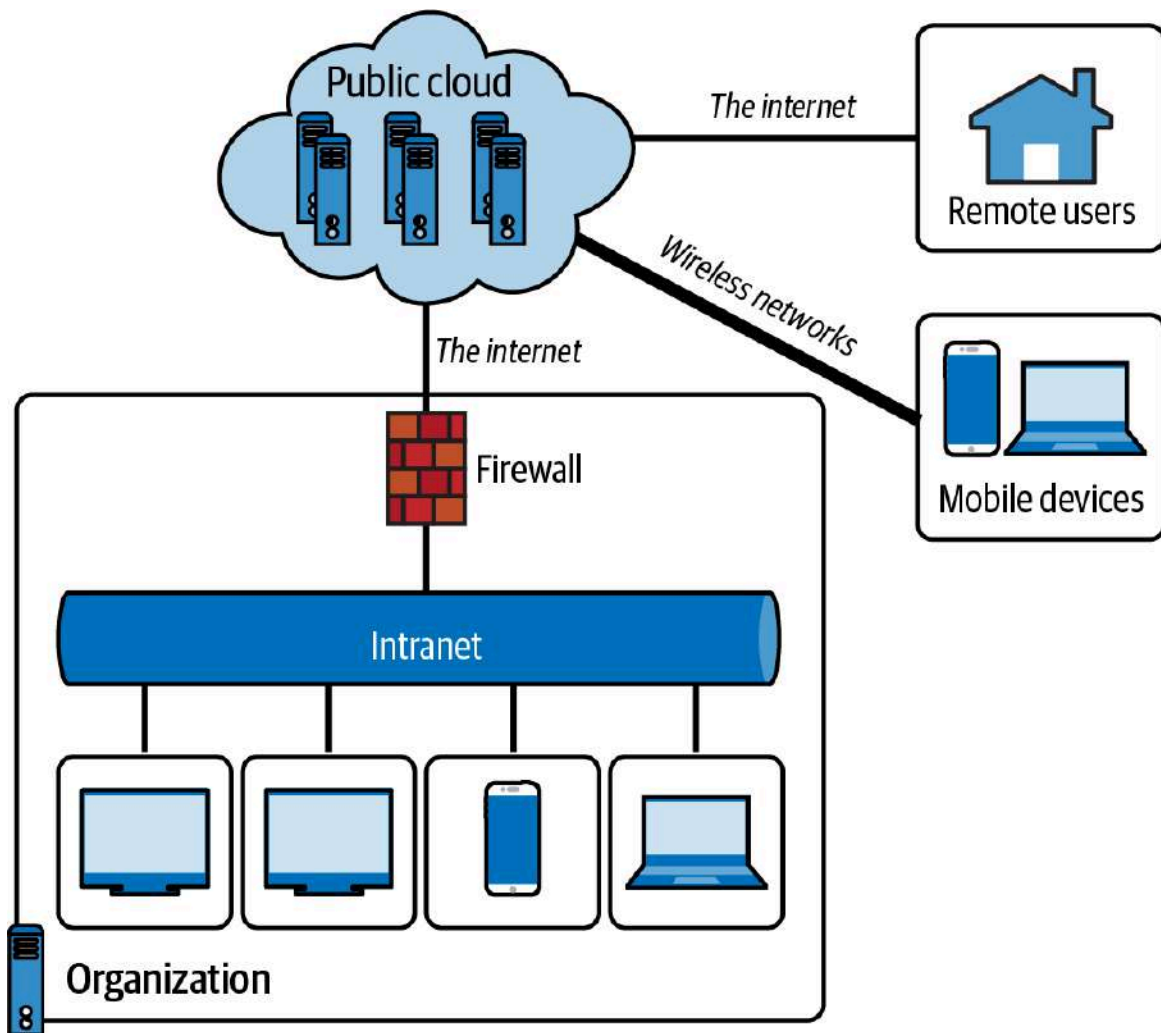


Figure 1-3. An example of several organizations using a public cloud

Advantages of using a public cloud

Whether your business is focused on cost reduction, global scale, or better administrative management, or it wants modern solutions with enhanced security, there are many great benefits to moving to the public cloud.

Services available by migrating to the cloud include:

- Cost effectiveness and cost management
- On-demand services and portability
- Scalability and reliability
- Sophisticated and modern solutions

- Flexibility in administration through self-service cloud management portals
- Monitoring, analytics, and report visualization
- Resource pooling
- Security and privacy
- Disaster recovery and geolocation

Private Cloud

A private cloud infrastructure is operated and owned by one organization: on premises or off premises. An organization utilizing private clouds uses cloud computing technology with considerations for privacy and security. This means that access to the resources in the IT infrastructure within the organization is centralized. Trust boundaries define the organization's administration of private clouds.

As shown in **Figure 1-4**, a private cloud infrastructure and its resources are typically managed on the organization's private cloud on a virtual private network. Organizations that use the private cloud in their IT infrastructure are responsible for managing and maintaining their infrastructure.

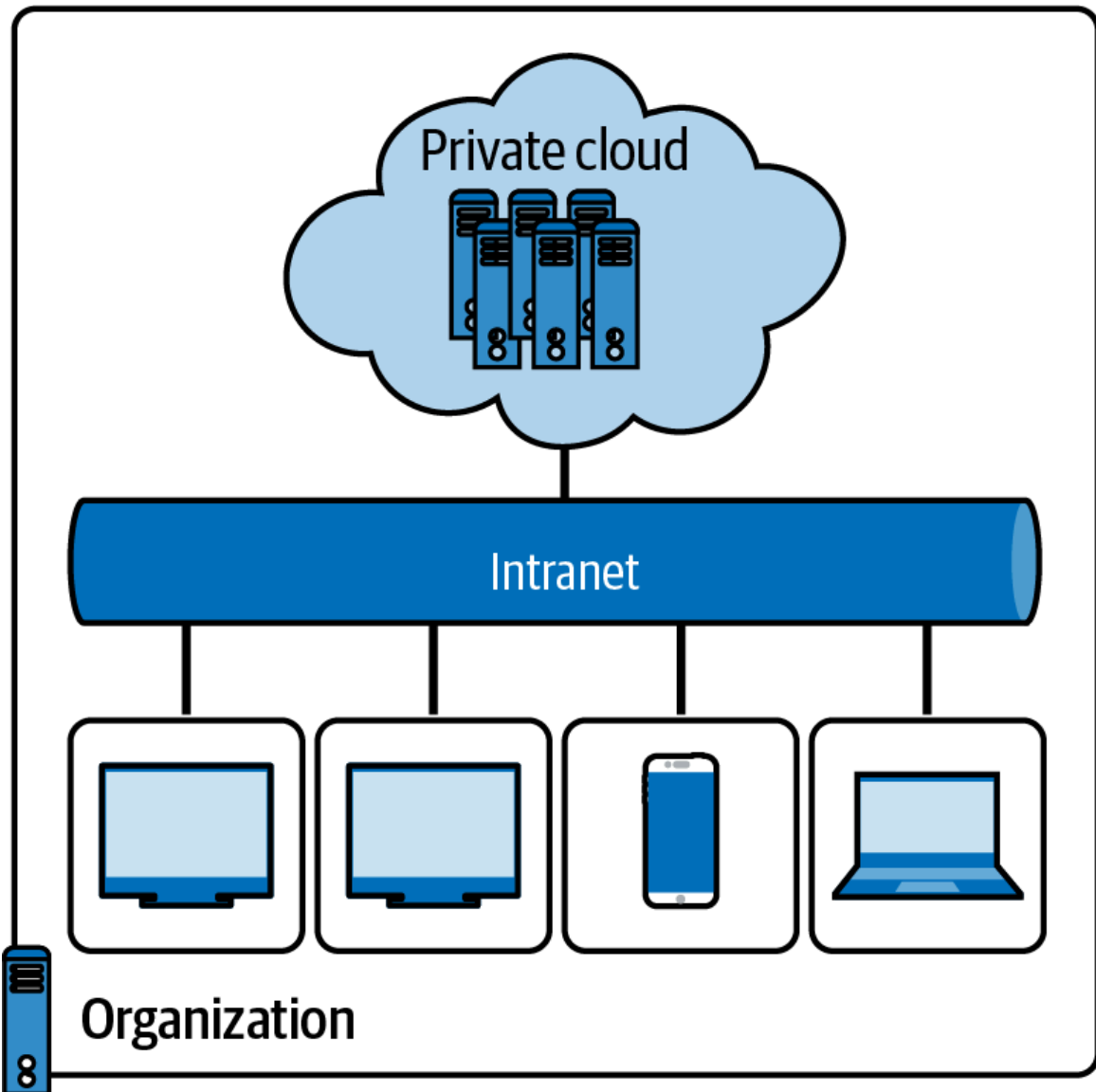


Figure 1-4. Private cloud example

Some institutions with special requirements and IT policies requiring enhanced security and control over the cloud infrastructure use private clouds. Financial institutions, government agencies, and organizations that require advanced security and strict privacy usually prefer this cloud.

Advantages of using a private cloud include:

- Enhanced privacy and security since resources are not shared with others
- Increased control over the infrastructure and owned resources

- Compliance with business-critical security and regulatory compliance requirements
- Flexibility to customize the environments based on the on-demand requirements of the organization or business

Community Cloud

A community cloud is a hybrid form of the private cloud. These multi-tenant platforms enable different organizations to work on a shared platform. This type of collaborative cloud is rarely mentioned publicly, but it is used widely.

Institutions and organizations that use community clouds include:

Government

Most cloud providers offer community clouds for governments; they are known as clouds for the government. For example, cloud provider Amazon AWS offers **Cloud Computing for Federal Government** while Microsoft has **Azure Government** for US government agencies and their partners. Clouds for the government sector are community clouds meant explicitly for the government since they have legal, security, and privacy regulations, processes, and services that require constant communication and data transactions between different departments. They all operate on the same infrastructure, with benefits and shared resources.

Healthcare

The US healthcare sector is regulated by **Health Insurance Portability and Accountability Act of 1996 (HIPAA) compliance**. This compliance holds and controls the security and transfer of sensitive patient record information such as medical records exchanged between hospitals and

laboratories. Healthcare sectors adopting the cloud use community cloud providers that adhere to HIPAA regulations. Aside from compliance, the healthcare sector is also adopting cloud computing technology to **improve healthcare services and costs using artificial intelligence and machine learning**.

Education

When **COVID-19** struck, it affected the education sector and institutions in many countries. Because of pandemic regulations, schools were challenged to deliver education in person. Internet, cloud computing, and remote access to education and learning materials have helped schools and universities develop online education for students. **Azure for Education** is a cloud provider that offers cloud services such as **Azure Lab Services, Teams**, Office 365 Educational Plans, and **Azure Virtual Desktop** for education.

Remote and hybrid work

Based on **recent statistics**, about 16% of companies globally are 100% remote while 77% of remote workers claim they are more productive when they are working from home. Cloud computing technologies are used for remote and hybrid work ³, and they will continue to change how we work and collaborate. Some companies enable remote employees to securely connect to their systems from any device over any network. Organizations also use community clouds for innovations governed by regulations before hosting in the public cloud. This means community clouds are being used as an initial setup hosting resources and infrastructure to a private cloud.

The infrastructure of the community cloud supports a specific community with shared missions, compliance, security, jurisdictions, etc. A community

cloud, as shown in **Figure 1-5**, can be managed by a community or organization that can be controlled or hosted internally or externally.

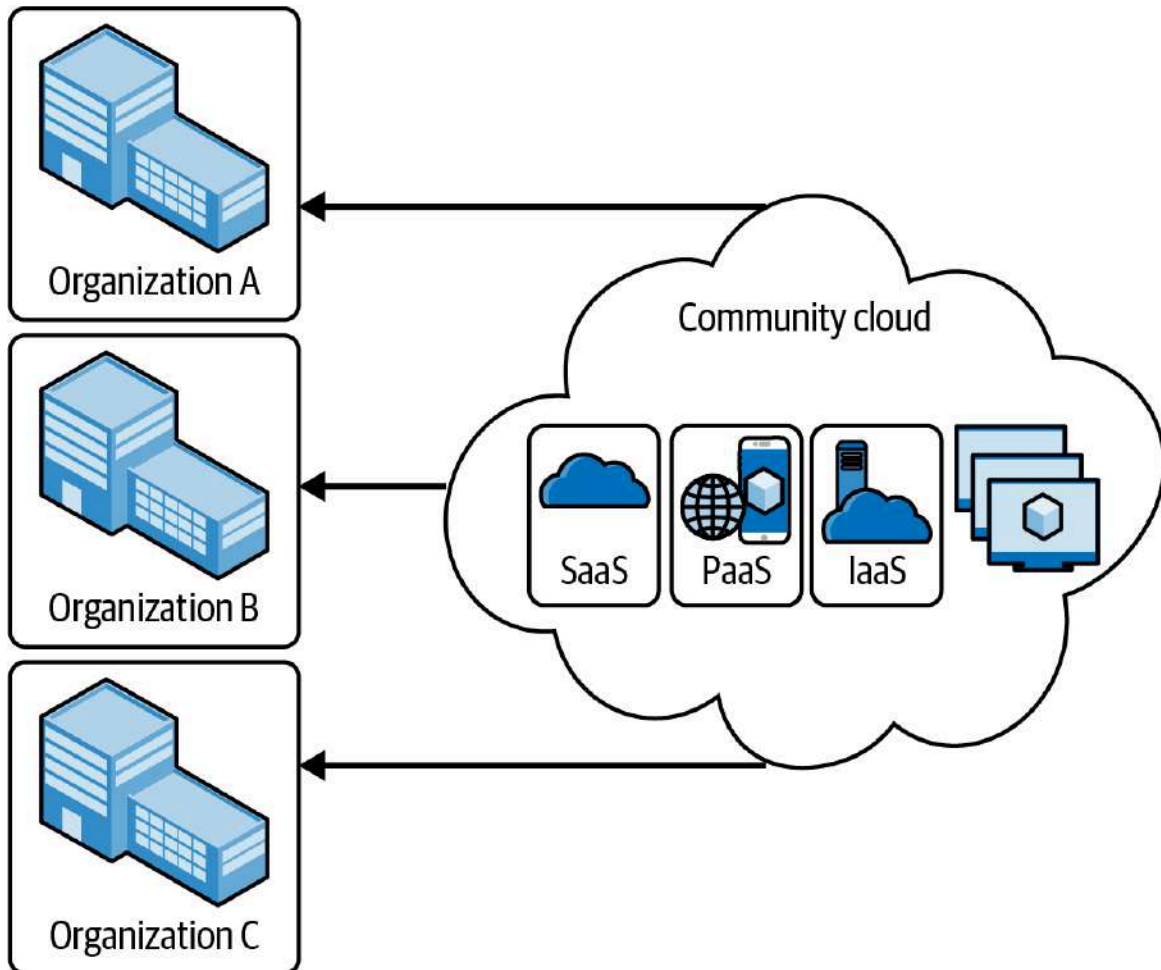


Figure 1-5. Community cloud

Hybrid Cloud

A hybrid cloud is a type of cloud infrastructure composed of multiple clouds, a combination of private, public, or community clouds, as shown in **Figure 1-6**. In hybrid clouds, unique entities are kept but bound together by standardized technology, which allows the portability of applications and data—for example, load-balancing between clouds through cloud bursting.

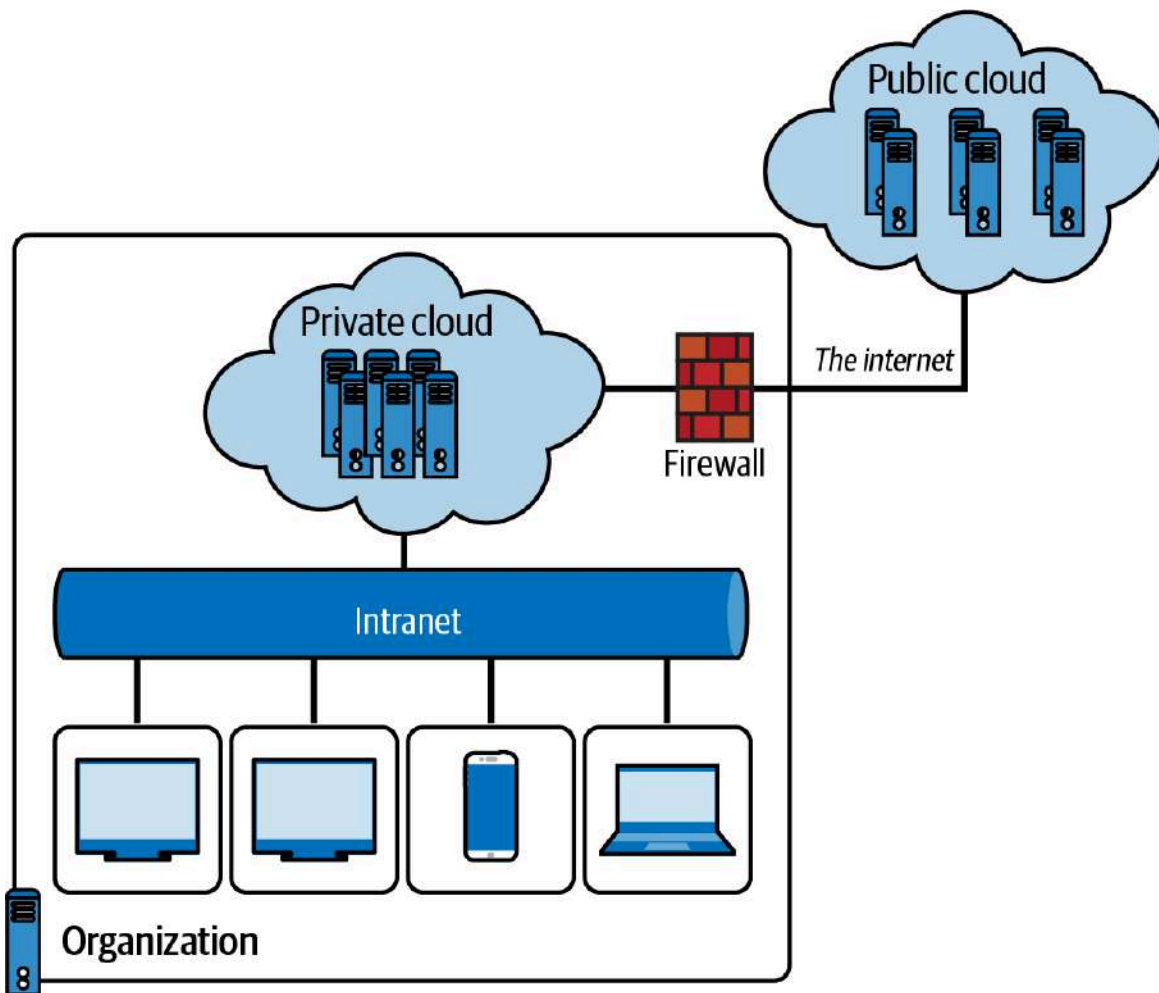


Figure 1-6. Hybrid cloud

Cloud bursting is common in hybrid cloud scenarios. It is an application deployment in which an application runs in an on-premises data center or private cloud. Then it can burst into a public cloud if the workload or computing capacity demands increase, thus granting access to more computing resources when needed.

Benefits of cloud bursting are its agility and ability to adjust to rapidly changing workloads. It also provides a cost-effective way to scale up and down. One practical example is the flexibility to handle compute workload issues by rerouting the traffic from a private cloud and expanding or “bursting” it to the public cloud.

What Is Multi-Cloud?

Multi-cloud refers to using multiple cloud computing services from different providers to meet a business's needs. In other words, rather than relying on a single cloud provider, an organization uses services from multiple providers, such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), or IBM Cloud.

Benefits of implementing a multi-cloud approach include:

Reduced risk of downtime

By using multiple cloud providers, organizations can reduce the risk of downtime or service interruptions. If one provider experiences an outage, the organization can quickly switch to another provider to keep its services running.

Cost optimization

Organizations can use different cloud providers for other purposes based on pricing, performance, and features. This can help optimize costs and prevent vendor lock-in.

Improved security

Multi-cloud can enhance security by reducing the risk of a single point of failure. Furthermore, organizations can use different providers for different security needs, such as those with specialized security features for sensitive data.

Flexibility

Multi-cloud helps organizations be more agile and flexible, enabling them to adapt to changes in the market and their business needs. For example, they can choose a provider based on their specific requirements for a particular project.

Better performance

Using multi-cloud strategy for IT infrastructure can help organizations achieve better performance by leveraging different providers' strengths and using the right provider for the right workload.

In summary, multi-cloud is a strategy that enables organizations to use multiple cloud providers' strengths to meet their business needs while improving flexibility, cost optimization, security, and performance.

Hybrid Cloud Versus Multi-Cloud

Hybrid and multi-cloud are two different cloud computing architectures, each with its own set of benefits, downsides, and **strategic considerations**. Some of the common reasons why organizations consider either of these two cloud infrastructure options include modernization, innovation, migration, and business requirements.

A hybrid cloud computing model combines public and private cloud resources to offer a comprehensive solution. This approach enables organizations to maintain control over their sensitive data by keeping it in their private cloud while also taking advantage of the scalability and cost-efficiency of public cloud resources.

Organizations can achieve greater flexibility in managing their workloads and data by using a hybrid cloud. They can use the private cloud for workloads requiring higher security and control levels. In contrast, the public cloud can be used for workloads that demand greater scalability and cost-effectiveness. A hybrid cloud approach offers a more adaptable and effective solution for organizations that require both security and flexibility in their cloud computing environment.

Benefits of a hybrid cloud include:

Improved security

Organizations can keep their most sensitive data in the private cloud and still leverage the scalability and cost-

efficiency of public cloud resources.

Greater flexibility

A hybrid cloud allows for more flexibility in managing workloads, as it allows organizations to choose the cloud environment that best suits their needs for each individual workload.

Cost efficiency

A hybrid cloud can reduce costs by allowing organizations to leverage the cost advantages of public cloud resources while keeping mission-critical data in the private cloud.

On the other hand, a multi-cloud architecture means the use of multiple public cloud providers for business strategy purposes. In this environment, an organization can use different cloud providers to meet different needs. For example, one provider might offer better storage solutions, while another might have better data analysis tools. Note that there are pros and cons when considering multi-cloud solutions in terms of storage.⁴

Common uses and benefits of multi-cloud include:

Avoiding vendor lock-in

Multi-cloud can help organizations avoid being tied to a single cloud provider and subject to their pricing and policies.

Increased resilience

Multiple cloud providers can provide additional redundancy and backup options to ensure business continuity and disaster recovery.

Agility

Multi-cloud allows organizations to choose the best cloud provider for each workload, ensuring that each workload has the best resources and capabilities available.

While hybrid and multi-cloud are different, both offer significant benefits in terms of flexibility, security, resilience, and cost-efficiency. The choice between the two depends on the specific needs and objectives of the organization.

Accordingly, it is important to understand the differences between these deployments so that you plan how you design your cloud architecture and infrastructure. Being aware of the **benefits and limitations of hybrid and multi-cloud** would be ideal for an organization's cloud strategy.

Public Cloud Computing Providers

This book is about learning Microsoft Azure; however, since we are learning about cloud computing and multicloud in this chapter, it is crucial to learn about the other public cloud providers in the market.

Migrating on-premises applications or systems to the cloud is a challenging process. It requires serious planning, strategy, and preparation.

It is difficult to say that one cloud provider is better than the other. However, choosing the appropriate cloud provider for your organization and your teams depends on your current IT infrastructure, the business problems you need to solve, and your organization's business motivations. Every cloud solution and its implementation should be aligned with the purposes and goals of a business.

Microsoft Azure

Azure is one of the fastest-growing cloud provider platforms offered by **Microsoft**. Even though Azure started years after its competitors, it is one of the leading global cloud computing providers.

Azure offers a wide variety of cloud services in different categories, including artificial intelligence, machine learning, analytics, blockchain, compute, containers, serverless computing, databases, developer tools, DevOps, identity management, integration, Internet of Things (IoT), edge computing, quantum computing solutions, cloud management and governance, media and communication services, Azure Hybrid, migration, mixed reality, mobile, networking, security, storage, web, and Windows Virtual Desktop.

What makes Azure one of the most attractive and intelligent solutions is its exclusive offering of Microsoft's products and integration of services in the cloud. Azure provides the most advanced and maximum number of smart products and services.

This book will cover Microsoft Azure in detail.

Amazon (AWS)

AWS is Amazon's cloud platform that offers various services such as Virtual Private Cloud, EC2, AWS Data Transfer, Simple Storage Service, DynamoDB, Elastic Compute Cloud, AWS Key Management Service, AmazonCloudWatch, Simple Notification Service, Relational Database Service, Route 53, Simple Queue Service, CloudTrail, and Simple Email Service.

It is one of the most broadly adopted cloud platforms. This cloud platform offers services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Google Cloud Platform

Google Cloud Platform (GCP) is Google's cloud and is also one of the top public cloud providers available. Similar to AWS and Microsoft Azure, GCP also offers services in various categories, including computing, storage, identity, security, database, AI and machine learning, virtualization,

DevOps, and more. Google Cloud Services are available in 20 regions, 61 zones, and 200+ countries.

GCP delivers a wide variety of IT products that IT professionals, businesses, and software developers can take advantage of to work more efficiently and gain more flexibility.

Oracle Cloud

Oracle Cloud Platform is the cloud offering of Oracle corporation. Oracle Cloud Platform offers infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), and data as a service (DaaS). Oracle SaaS offerings are Oracle Cloud CX, Human Capital Management (HCM), Enterprise Resource Planning (ERP), Supply Chain Management, EPM, IoT, analytics, data, and blockchain applications. Oracle DaaS is the Oracle Data Cloud.

Alibaba Cloud

Alibaba Cloud, founded in 2009, is lesser known in some parts of the world but is also a prominent public cloud provider. It is the largest cloud provider in China. Alibaba is registered and headquartered in Singapore, and it was initially built to serve Alibaba's e-commerce ecosystem.

They offer various products and services in multiple categories, including elastic computing, storage and CDN, networking, database services, security, monitoring and management, domains and websites, analytics and data technology, application services, media services, middleware, cloud communication, Apsara Stack, and IoT.

If you want to learn more about how these different public cloud providers are leading, I recommend [**Gartner: Magic Quadrant for Cloud Infrastructure and Platform Services**](#).

Cloud Computing Service Models

There are different cloud service models in the cloud. These are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), as shown in **Figure 1-7**.

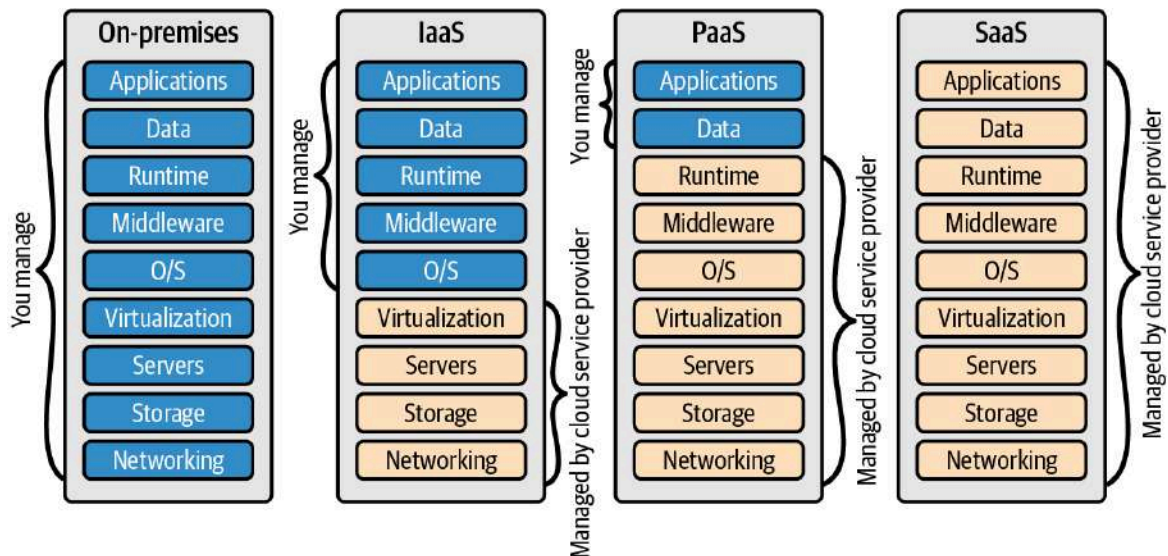


Figure 1-7. The different cloud computing models show what is managed by the cloud provider and what is managed by the cloud user or organization

Infrastructure as a Service

Infrastructure as a service (IaaS) is a computing deployment model category in which the public cloud provider like Azure delivers infrastructure through the cloud. Instead of the traditional infrastructure on premises in one physical location, the services are provided by public cloud vendors.

IaaS is a delivery of IT infrastructure resources like web servers, database servers, compute storage, networking, computing data centers, and other compute resources available as a service and on demand. When a cloud computing user uses the IaaS deployment model, they don't need to worry about maintenance costs or the hassles of having infrastructure on premises in the traditional structure.

Platform as a Service

Another cloud deployment model category commonly used is platform as a service (PaaS). It is a cloud model where users, organizations, or developers can create, build, and deploy applications on the cloud without worrying about the IT infrastructure behind it. This means that Microsoft Azure, as one of the PaaS cloud providers, typically provides a range of computing services, development, and monitoring tools for application development on the cloud.

In this cloud model, typically, Azure is responsible for taking care of the physical infrastructures, data centers, hardware, operating system, middleware, and other resources required to run and manage the application. The developer is only responsible for writing and deploying the application code on the platform.

Software as a Service

A few examples of standard software as a service (SaaS) offerings are Salesforce, Microsoft 365, Dropbox, OneDrive, Google Workspace and other applications offered as services.

The SaaS model enables users to access and use applications online with sync capabilities. It is more flexible and convenient than installing and running the applications on their computers or servers. Users can access the applications through a web browser or other client software and typically pay for the service on a subscription basis. SaaS applications can be used for various purposes, including productivity tools, customer relationship management, and enterprise resource planning.

SaaS is a software on-demand cloud model where the cloud service providers give the users access to a fully developed application explicitly created for distribution. The software updates are rolled out for all users, and organizations can use their tools with vendor-provided application programming interfaces (APIs).

One way to compare the cloud computing service models in the real world is to illustrate using our favorite food, pizza. Consider the pizza example in **Figure 1-8**, which was originally created in 2014 by Albert Barron, who used to work as a Software Client Architect at IBM. ⁵

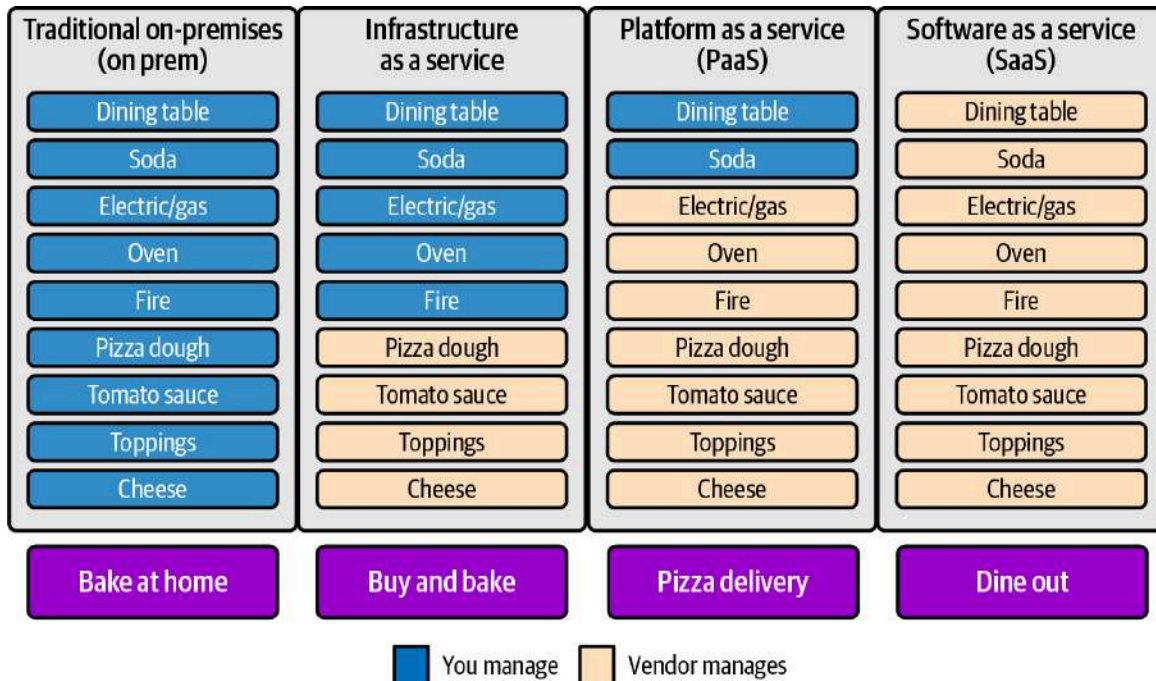


Figure 1-8. An example of cloud computing as pizza as a service (adapted from an image by “*Pizza as a Service*”)

There are similarities between baking our pizza and having our traditional on-premises IT infrastructure. Shopping for a pizza from the store and baking it at home is like IaaS. Pizza delivery is like PaaS, and dining out at a restaurant to order pizza is like SaaS offered by the cloud provider.

Aside from IaaS, PaaS, and SaaS, other cloud computing service models are available. These additional service models are serverless, function as a service (FaaS), backend as a service (BaaS), and more.

Serverless Computing: Function as a Service and Backend as a Service

Serverless, FaaS, and BaaS are terms that have gained popularity and interest in the cloud computing industry. Serverless is a method of

computing where backend services are provided by a cloud service provider.

In serverless computing, a third-party provider manages the infrastructure and automatically provisions and scales resources as needed, allowing developers to focus solely on writing code for their applications.

In a serverless computing model, the provider handles the server infrastructure, operating system, and other lower-level components while developers provide the application code. This approach can significantly simplify the development process by allowing developers to focus on application logic and functionality without the need to manage the underlying infrastructure.

Serverless computing is related to BaaS and FaaS. They all provide an abstraction layer between the developer and the underlying infrastructure. However, they differ regarding the level of abstraction and the services provided.

Technically, “less” in this term means the servers and underlying infrastructure are abstracted. There are servers behind a serverless function or serverless cloud service, but the cloud provider or serverless provider is taking it off them for its users. Cloud services on serverless usually have consumption pricing models where the users are charged on usage and execution.

FaaS is a technical concept that aims to allow developers the freedom and productivity to easily create functions in a cloud environment. In this type of architecture, the developers will still create the application logic, yet the code is executed in stateless compute instances managed by the cloud provider. FaaS provides an event-driven computing architecture where a specific event, such as message queues, HTTP requests, etc., can trigger a function. In Azure, different serverless solutions, such as Azure serverless compute services for applications and serverless containers, are available. These compute services for serverless or event-driven applications such as Azure Functions, Azure Logic Apps, Azure Event Grid, Azure Event Hubs, and other services are discussed in detail in [Chapter 3](#).

Azure compute services like Azure Functions, as shown in **Figure 1-9**, allow users to build applications faster by eliminating the hassles of managing servers and infrastructure. It enables software developers or programmers to focus on the productivity of their development teams and pay only when the code runs. Developers can focus on developing event-driven applications using their chosen supported language.



Figure 1-9. Microsoft Azure Functions as function as a service (FaaS)

Implementing serverless and FaaS solutions has several benefits, especially for the software development team. In software engineering, we developers want to focus on delivering solutions, solving problems, and building applications.

Developers and engineers do not want to spend time maintaining servers and infrastructures. Usually, they prefer to deliver value and solutions by programming and developing applications whether they are on premises or on the cloud.

With serverless computing, FaaS, or **BaaS**, developers or cloud engineers can focus more on productivity by focusing on the backend logic and not worrying about infrastructure management. This results in speed of delivery, which helps the project process.

Aside from developer benefits such as speed of delivery, the opportunity to solve problems with complex applications is available. Other benefits include automatic scaling, reliability, and a consumption-based pricing model.

Serverless architecture is a crucial software architecture design pattern that partly relates to **distributed computing systems** and **microservices**. Developers need not worry about managing and maintaining infrastructure,

hardware, or servers but can focus more on developing logic and functionality. They write the code and leverage the infrastructure of cloud provider services and other third-party services, or BaaS. We will cover serverless and compute solutions in Azure in [Chapter 3](#).

Containers as a Service

Containers as a service (CaaS) or container development is an exciting solution. By utilizing containers, you get PaaS benefits without the overhead of IaaS.

Containerization, in simple terms, is deploying your applications into the container. A container is a runtime that contains the essential computing resources needed to run an application. This includes the core part of the host operating system (also known as a kernel) and its shared resources like storage across a host. The shared kernel allows containers to be lightweight and faster.

When hosts are running, the containers in them can start quickly. Quickstarts mean high availability and resiliency of the applications in the container. One example of containerization technology is Docker, one of the more popular providers of container services. Compared to traditional virtual machines, containers:

- Can run in cross-platform environments
- Are lightweight and portable
- Are self-contained with no need to install application dependencies
- Have good scalability and high availability
- Are quick to restart

Container development and other compute services in Azure will be covered in the [Chapter 3](#).

Data as a Service

Every website, application, system, mobile app, and tech product we use has data in it. Sensitive data is being protected through data protection policies.

Data as a service (DaaS) focuses on providing data as a business asset by implementing data management strategies. This cloud service model gives organizations better agility.

As listed in **Table 1-1**, DaaS provides organizations effective strategies on how to handle, manage, and visualize the massive data that is generated every day.

Table 1-1. Benefits of data as a service (DaaS) as a cloud service model

Benefit	How?
Data-driven culture	DaaS enables organizations to organize and manage their increasing data by using datasets that are reusable and easier to analyze and visualize.
Innovation and business growth	DaaS puts data as a critical driver in the business, opening growth and innovation opportunities. Data-driven strategies drive innovation and growth without creating huge risks.
Scalability, reliability, and flexibility	Cloud solutions usually offer DaaS solutions that are flexible and scalable.
Data monetization	Solving data operation problems and complexity can also be beneficial in monetizing the valuable data.
Cost savings	DaaS solutions can help organizations save expenses by allocating the appropriate workloads for their data in the cloud.

The challenges of data as a service

Although DaaS offers great benefits to an organization, it has some known challenges:

- Risks of solving data complexity problems especially for old and unstructured data sets

- Implementing a data-driven culture with DaaS requires an top-down organizational and business strategy
- Higher demand for management of data privacy and security because of different data privacy regulations and compliance requirements

Shared Responsibility in Cloud Computing and Azure

Considering and evaluating cloud services on any public cloud provider requires careful planning and strategy. It is critical to learn and understand the shared responsibility model, which helps both parties (the user and cloud vendor) share the trust and responsibility of hosting applications and resources in the cloud.

For example, it helps identify which security tasks are handled by the cloud provider and which jobs are handled by the public cloud user. The shared responsibility for the workload varies depending on whether the workload is hosted on SaaS, PaaS, or IaaS, or in an on-premises data center.

With Microsoft's example of the **Cloud-Enabled Shared Responsibility Model**, your organization will be able to properly choose the right services and deployment model for your use case. The model also gives you a good overview on which components of your infrastructure are your responsibility as an organization, and which ones are the responsibility of the cloud provider, Microsoft Azure.

Shared Responsibility Model Offers Cloud Security Advantages

The cloud offers significant advantages for solving long-standing information security challenges. In an on-premises environment, organizations likely have unmet responsibilities and limited resources available to invest in security, creating an environment where attackers can exploit vulnerabilities at all layers.

As a security framework commonly used by organizations and cloud providers, the shared responsibility model addresses the security faults of cloud service providers and their users. This security framework is an excellent model to follow because it guides them through how they would architect, design, develop, process, and manage their data through their users and applications.

For example, in this type of model focusing on security, the cloud service provider is responsible for the protection of the cloud infrastructure like data centers, which usually includes ensuring the infrastructure is always available and keeps its integrity, along with confidentiality, reliability, and other factors in terms of compliance with local regulations within that specific region.

On the other hand, the client or user of the cloud services is responsible for securing their data and applications stored and used in the cloud. This includes implementing appropriate access controls, managing user identities and authentication, and securing data in transit and at rest.

The cloud service provider, Azure, and the users have a shared responsibility for security, with each party responsible for different aspects of the cloud environment's safety. All of the parties involved must understand their responsibilities to ensure the cloud environment is secure and compliant with relevant regulations.

Figure 1-10 shows a traditional approach in which many security responsibilities are unmet due to limited resources. In the cloud-enabled practice, you can shift day-to-day security responsibilities to your cloud provider and reallocate your resources.

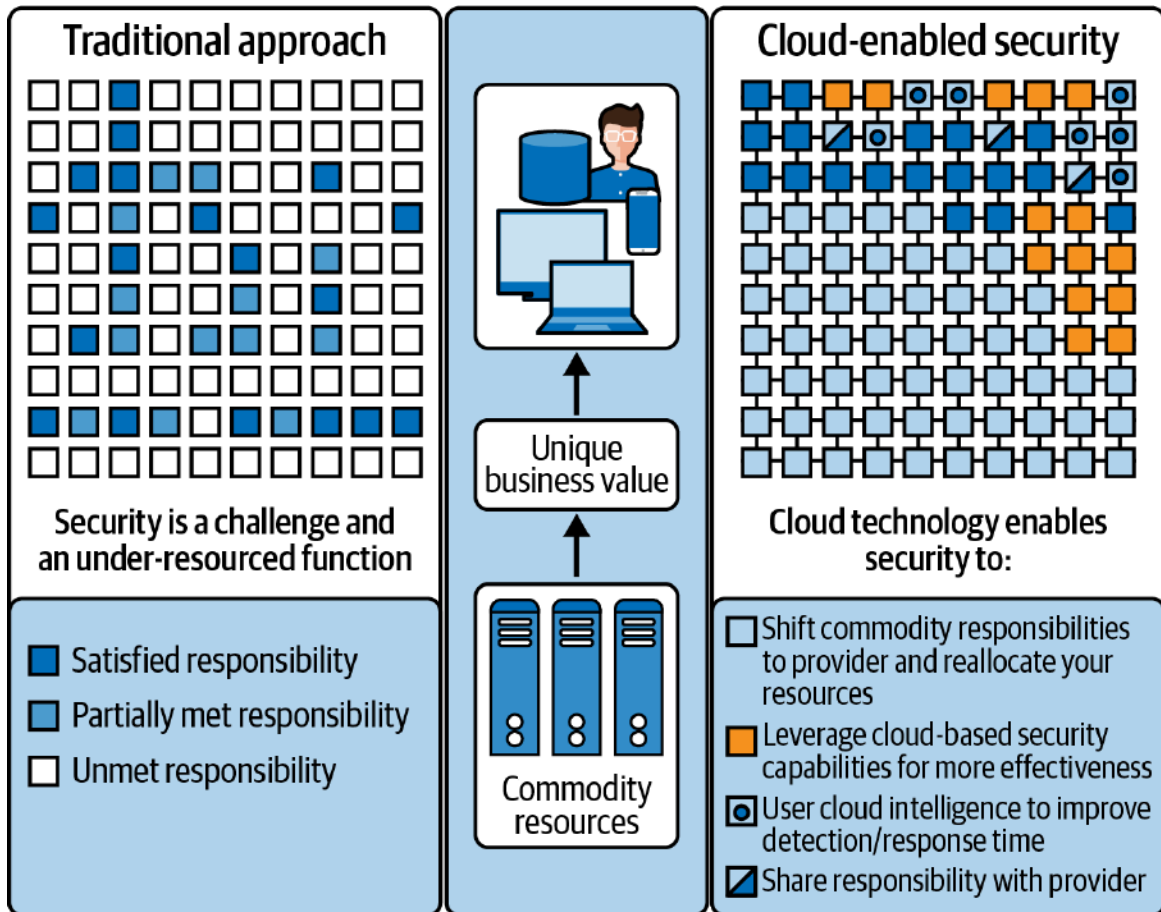


Figure 1-10. Microsoft's cloud-enabled Shared Responsibility Model

Capital Expenditures and Operational Expenditures

An organization can't migrate on-premises resources to the cloud without knowing about the benefits for capital expenditures (CapEx) and operational expenditures (OpEx). Comparing these two when considering cloud computing solutions for businesses or organizations is vital.

Capital expenditures (CapEx)

CapEx refers to capital costs or expenses required for cloud computing. Usually these expenses involve physical assets such as buildings needed for the IT infrastructure, networking equipment, data centers, human resources, and

other resources to get started. These types of capital expenses are typically fixed and are not flexible. They cannot be adjusted based on usage. For example, if a business purchases a new web or database server, it will incur a one-time cost that cannot be dynamically adjusted or scaled based on the level of usage of its users or the entire organization.

Operational expenditures (OpEx)

OpEx refers to expenses that are incurred on an ongoing basis to keep the business running, such as cloud computing services, software licenses, salaries, and maintenance costs. These expenses are typically variable and can be adjusted depending on the level of usage, making them more flexible than CapEx. With the adoption of cloud computing, businesses can reduce their CapEx and shift toward OpEx, as they can pay for the services they use on a pay-as-you-go basis, allowing for more efficient resource allocation and cost savings.

Typically, organizations start with traditional on-premises physical servers and data centers that require expensive CapEx. Cloud computing solutions offer organizations the option to eliminate the hassles of traditional infrastructure on premises by providing services with OpEx alternatives.

Benefits of Adopting and Transformation to Modern Cloud Technologies

Cloud adaptation and transformation is a complex and lengthy process. It is not a quick-fix solution to modernize on-premise and legacy applications quickly to the cloud. However, when planned adequately with intelligent strategies, there is an excellent list of benefits.

Cloud Computing for Business Value and Customers

How does an organization or business benefit from cloud computing? Regardless of the size of the business or organization, cloud computing helps in saving resources, time, and money by accelerating innovation, collaboration, modernization, and productivity in different teams within the organization. It also helps provide business value for users because of the enhanced user experience, speed, and reliability of modern cloud applications.

Cloud Computing for IT Companies

IT organizations gain many benefits from using cloud computing solutions:

Data access management and portability

Cloud computing lets businesses portably access their important business-related data anywhere on any device. This capability allows the entire organization to work effectively and productively by focusing on deliverables. With cloud storage and servers on the cloud, employees in the organization can be out of the office and access the intranet to work and collaborate. Information is available securely at any time on demand.

Cost management and efficiency

Buying and maintaining server equipment requires time, expertise, and money; rather than building your bespoke server, which can be prone to downtime, a cloud computing provider stores data for you without all the downsides. Prices for business-oriented cloud services are still monthly, but it's a manageable and predictable expense.

Convenient backup and disaster recovery solutions

Catastrophic data loss can happen at any time, and it can be time-consuming to solve such a significant issue when it

happens. Whether that loss occurs from natural disasters, power surges, or hardware failure, affected companies are at increased risk of bankruptcy within the same year as the data loss. And while most companies have adopted backup plans, it helps to have additional contingencies. By utilizing the cloud to store important data, business owners can rest easy knowing that important files are safe even if hardware fails. For cloud computing solutions in Microsoft, a great variety of backup and disaster recovery options are available for applications hosted in Azure.⁶

High-level cloud security and data privacy

Security and privacy are critical in the decision-making process when using cloud computing services. The public cloud computing provider and the user or organization have shared responsibility. Hosting applications and servers on the cloud is built on trust. This is the reason for setting clear expectations and being familiar with the shared responsibility model. Cloud providers prioritize security and data privacy for their clients and consumers. They use different strategic cloud security controls to protect their users' resources. Identity management, high-level physical security in data centers, strict personal protection, and ensuring data privacy for sensitive data are taken seriously. Dedicated security expert teams scan the cloud for possible internal and external vulnerabilities. Performing cloud penetration testing and examining inside and outside the cloud must be strictly authorized. Cloud providers do not tolerate any security breaches. Services are terminated when security and use policies are violated. Cloud computing impacts our daily routines at work, home, and school. It has improved our facilities for healthcare, education, and communities, and it is continually changing our digitalization worldwide.

Digitalization and modernization

Digitalization and modernization of many computer systems in different sectors was made possible because of the advancement of technologies hosted on the cloud over the internet. Institutions in sectors like healthcare, government, and education are using the cloud to modernize their services and products.

Remote and flexible education and digital literacy

Aside from digitalization and modernization, cloud computing also improves education worldwide. It activates the capability and capacity of online collaborative and self-paced modern learning environments.⁷ Remote locations with internet access facilities give opportunities for literacy through online education.

Summary

In this chapter, you have learned about the fundamentals of cloud computing, its different types, and its deployment models. You also learned about the history and evolution of cloud computing that helped shape the modern cloud.

You saw an overview of the different types of clouds (public, private, hybrid, community, and multicloud) and know the differences between them. The other deployment models, such as IaaS, PaaS, SaaS, CaaS, DaaS, FaaS, and serverless, are helpful when choosing which cloud service to use in a cloud platform.

You learned insights and the advantages and benefits of cloud computing in businesses, IT companies, our society worldwide, and software engineering.

In the next chapter, we will learn more about the vital concepts you need to know about Microsoft Azure as a cloud platform. We will learn what cloud

solutions Azure has to offer to help you and your organization.

Check Your Knowledge

1. What is cloud computing? Why is it important to society today?
2. What are the differences between the public cloud and the private cloud? Which would you consider using for your organization?
3. Reflect on and explain the importance of understanding the shared responsibility model security framework when using cloud services such as Microsoft Azure.
4. What is the difference between hybrid cloud and multi-cloud?
5. What is CapEx and OpEx?

For the answers to these questions, see the [Appendix](#).

Recommended Resources

“Cloud Computing Basics for Developers.” Microsoft Learn,

<https://oreil.ly/YKN3o>.

“Cloud Computing Platform Market by Service Model, by Deployment Model, Organization Size, Vertical and Region - Global Forecast to 2027.”

Reportlinker.com, <https://oreil.ly/qHA2H>.

Foote, Keith D. “A Brief History of Cloud Computing.” Dataversity.net,

May 4, 2023. <https://oreil.ly/FcpoQ>.

“Mainframes and Cloud Computing: Similarities and Differences.”

Turbosoft, May 19, 2015, <https://oreil.ly/eU3bC>.

Regalado, Antonio. “Who Coined ‘Cloud Computing’?” MIT Technology

Review, October 31, 2011, <https://oreil.ly/kDbM1>.

“What Is Cloud Computing? A Beginner’s Guide.” Microsoft Azure documentation, <https://oreil.ly/1Gh-2>.

“What Is the Cloud?” Microsoft Azure documentation, <https://oreil.ly/ekOql>.

-
- ¹ Kief Morris, 2020, *Infrastructure as Code*, O’Reilly Media, <https://learning.oreilly.com/library/view/infrastructure-as-code/9781098114664>
 - ² Bernard Marr, October 25, 2021, “The 5 Biggest Cloud Computing Trends in 2022,” Forbes.com, <https://www.forbes.com/sites/bernardmarr/2021/10/25/the-5-biggest-cloud-computing-trends-in-2022/>
 - ³ Shardul Bhatt, 2021, “Using Cloud Computing to Build a Remote Organization,” Business2Community.com, <https://www.business2community.com/cloud-computing/using-cloud-computing-to-build-a-remote-organization-02382933>
 - ⁴ Rene Millman, “Multicloud Storage 101: Pros, Cons, Pitfalls, and Strategies,” ComputerWeekly.com, <https://www.computerweekly.com/feature/Multicloud-storage-101-Pros-cons-pitfalls-and-strategies>
 - ⁵ Albert Barron, “Pizza as a Service,” <https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service>
 - ⁶ Microsoft documentation, “Backup and Disaster Recovery for Azure Applications,” <https://docs.microsoft.com/en-us/azure/architecture/framework/resiliency/backup-and-recovery>
 - ⁷ James Riddle, “Cloud Technologies in the Education System,” IEEE Computer Society, <https://www.computer.org/publications/tech-news/build-your-career/cloud-technologies-in-the-education-system>

Chapter 2. Microsoft Azure Fundamentals

Cloud computing has revolutionized the way businesses run their IT workloads. It allows organizations to not only build and deploy applications in a modern way by using modern PaaS/SaaS services but also makes use of cloud computing advantages like high availability, scalability, security, and cost optimization. Understanding the Fundamentals is critical to building an excellent foundational knowledge of Microsoft Azure that later you can use to build and manage complex workloads. If you are just getting started in your cloud career or if you need to get deeper knowledge on Microsoft Azure, get your fundamentals right!

— Tiago Costa, Cloud Architect and Advisor, Microsoft
Azure MVP, Microsoft Certified Trainer

In **Chapter 1**, we learned about the important concepts of cloud computing; organizations use cloud computing resources with flexible pricing models to take advantage of the global scale, reliability, and security they provide.

In this chapter, you will enhance your knowledge of cloud computing by digging deeper into the world of Microsoft's cloud computing platform, Azure. You will learn the core concepts of Azure as a cloud provider: how Azure started, its purpose, and how it can inspire you to be innovative as you build modern solutions for your organization's demands and requirements.

You will learn about *Azure geographies, regions, and availability zones* as well as *Azure Resource Manager*, which gives you an overview of Azure's cloud infrastructure.

This chapter will give you a comprehensive overview of the core services in Azure. Each of those core Azure services will be explained in detail with

uses and examples in the upcoming chapters of this book.

NOTE

The public cloud is a type of deployment model where the cloud infrastructure is available to the public or any organization, and the resources are shared with other organizations or other cloud provider tenants.

Microsoft Azure as a Public Cloud Provider

Microsoft's computing platform is *Microsoft Azure*, also known as Azure, which is a sky blue color and the typical color of “the cloud.” Historically, Windows Azure was originally built on Windows Server 2008 for developers who wanted to host their software applications in a Windows environment on the cloud.¹

NOTE

Windows Azure became Microsoft's foundation for the cloud. It became Microsoft's great and powerful cloud platform. In 2014, it was renamed Microsoft Azure. Originally, Azure used the code name *Red Dog*.

Azure's ability to create, build, deploy, and manage organizations' applications on a global scale made it one of the top cloud providers worldwide. Microsoft Azure has grown to become a public cloud provider used by **95% of Fortune 500 companies**.

Microsoft Azure Helps Organizations Minimize Up-front Costs

The public cloud has features that are useful to many. These features include not requiring up-front costs for capital expenditures (CapEx) to

scale up resources, quick provisioning and deprovisioning of applications, and flexibility for organizations to only pay for what they use.

Azure is a public cloud provider, but it also offers private, hybrid, and multi-cloud solutions to users. Cloud services in Azure are designed to help users build innovative cloud solutions that help solve our challenges.

It allows us to build, develop, manage, and run resources like servers, databases, storage, or applications in multiple cloud environments. You can use it for different use cases in the cloud using the tools, programming languages, and frameworks that you prefer to use.

Azure offers over 200 services (as of time of writing) in various categories, including computing, networking, storage, databases, analytics, machine learning and AI, the Internet of Things (IoT), cloud-native, containers, and security. However, Microsoft continues to introduce new features and services to Azure regularly, so the number of services may increase over time.

Azure also provides edge computing, AI, and machine learning (ML) services that enable you to create intelligent solutions with your existing or new applications. Without the technology of cloud computing, this would not be possible.

Benefits of a Cloud Provider

There are several benefits that a cloud provider has over the traditional on-premises physical environment, as shown on [Table 2-1](#).

Table 2-1. Benefits of cloud versus on premises

Benefit	Description
High availability	Microsoft Azure provides high availability and redundancy across all of its worldwide data centers offering a service-level agreement that ensures 99.95% availability.
Geo-distribution	Azure helps global enterprises by providing geo-distribution features. Geography-specific endpoints enables international enterprises to comply with regional compliance and regulations.
Scalability on-demand	When demand for complexity, traffic, and data expands, there is a flexible and quick way to handle such needs
Reliability	The system or application hosted functions correctly even in the face of adversity (hardware or software faults, and even human error)
Elasticity	A capability to automatically scale cloud resources based on configuration or demand.
Disaster recovery	When your applications, data, and systems are hosted in Azure, you can be assured of secured end-to-end backup and disaster recovery solutions.
Flexibility	Cloud services in Azure gives organizations flexibility by allowing them to use consumption pricing plans and full self-service management accessible anywhere.

Benefit	Description
Cost management tools	Tools available for cost management in Azure allow users to set budget alerts for their resource groups and resources.
Advanced compliance	Azure takes information security and compliance seriously with high standards to serve its users.
High-level cloud security	Azure protects all stored data with world-class, advanced security. Data stored in Azure is protected with advanced encryption technologies. All Microsoft data centers are equipped with extreme security with biometric scanners, multi-level authentication and more.
OpEx vs. CapEx	Hosting to cloud means an organization can save money from capital expenditures (CapEx) and only pay for operational expenditures (OpEx).
Consumption-based pricing model	Azure gives its users the flexibility of cost management by offering a consumption-based (<i>pay-as-you-go</i>) pricing model in most of its cloud services.
No deep technical skills required	You and your organization do not need to be very skilled technically to get started using the cloud platform. Azure provides flexible and diverse options that make it easy to use.

Azure Portal

The self-managed portal of Microsoft's cloud platform is called the *Azure Portal*; it can be accessed by Azure users on their web browsers or via the *Azure mobile app*. It is a web-based administration website for all types of Azure users, where you can manage the Azure cloud services for your organization. It is a powerful portal of cloud administration tools and resources. For example, in Azure Portal, you can manage your resource groups, the resources in them, Azure subscriptions, security, monitoring, and more.

TIP

- Access Microsoft Azure Portal using the Azure mobile app. You may download it using the instructions in the [Microsoft Azure documentation](#)
- Access [Microsoft Azure Portal](#) using your favorite web browser on any devices.

The Azure Portal is built to be portable and accessible from anywhere on any device. [Figure 2-1](#) shows the user interface of the Azure mobile app, which you can use to manage your Azure subscriptions and services on the go.

TIP

The URL of the Azure Portal is specific to the cloud where your organization is deployed in Microsoft Azure.

- For commercial use or Azure Public Cloud:

<https://portal.azure.com>

- For Azure United States Government Cloud:

<https://portal.azure.us>

- Azure Germany:

<https://portal.microsoftazure.de>

- Azure China:

<https://portal.azure.cn>

Aside from the different locations and different scope of Azure as a cloud platform in the government sector and different countries, there is also **Microsoft Cloud for Sovereignty**. This is a solution that will help Azure users in government and public sectors leverage the cloud based on their specific and unique requirements. It will help these sectors gain better control over their data, privacy, compliance, and governance needs.



Home

Azure services



Virtual machines



Web Apps



SQL databases



Application Insights

Favorites

[See All](#)



Subscription
Pay-As-You-Go



Subscription
Pay-As-You-Go Dev/Test



Service Health



Resource groups



List

Chart

Latest alerts

[See All](#)



Home



Subscriptions



Resources



Notifications



Cloud Shell

Figure 2-1. Start page of Azure mobile app

Features of Azure Portal

The Azure Portal has numerous features for all types of Azure users. Organizations can take control of cloud resources by governing their cloud resources on-demand globally. Software developers or cloud engineers can build, manage, and monitor any type of cloud application, from simple to complex, regardless of architecture or programming languages. The following are some features of the Azure Portal:

- Create, build, manage, and monitor Azure services and cloud resources all in one place anytime and anywhere at your own convenience
- Use command-line tools and the cloud shell for quick creation and deployments
- Manage and organize Azure subscriptions and create management groups that helps in structuring and governing Azure resources
- Use Microsoft Entra ID to manage identity, access, and permissions to resources in Azure
- Configure and manage privacy, data, security, policies, and compliance vital to the organization's governance
- Customize the portal's dashboards to get a quick overview of the status of resources right after you log in
- Take control of monthly costs by monitoring resources through spending limits and budget alerts using Azure Cost Management in the Azure Portal
- Search everything you need to create, build, and manage using the **Global Search** feature in the portal
- Send **Azure Support requests** directly when you need assistance or help

TIP

Azure Marketplace is a marketplace for Azure customers to search, purchase, and try out applications and services from other service providers including Microsoft partner companies. All services on Azure Marketplaces are verified and certified to work with the Azure cloud platform.

Microsoft Azure Services

As of this writing, all Azure services are divided into **21 categories** according to their purpose. As a developer or a solution architect in an organization, or even as a beginner, it might be overwhelming to see a lot of cloud services in a cloud platform such as Azure. However, each Azure service has its unique purpose and is built to solve specific technical problems. Azure services can be seamlessly integrated with other services.

Each category for Azure services helps you build and integrate cloud solutions based on your business requirements or needs. For example, building a web application with integrations of APIs, cognitive services, and reporting features would need several Azure services: artificial intelligence/machine learning, compute, analytics, databases, integrations, developer tools, etc.

Table 2-2. Overview of Microsoft Azure service categories

Category	What are these categories?	Azure services
Artificial Intelligence (AI) + Machine Learning (ML)	Build modern cloud apps with ML and cognitive integration	Azure Boot Service, Azure Cognitive Services, Azure Machine Learning, Azure AI Anomaly Detector, Azure DataBricks, Azure Open Datasets, Computer Vision, Face API, Azure AI Immersive Reader, Azure AI Document Intelligence, Kinect DK, Microsoft Genomics, Azure Health Boot, Azure Applied AI Services, Azure Percept, Speech Services, etc.
Analytics	Gather and visualize any type of data regardless of its volume or velocity	Azure Analysis Services, Azure Data Explorer, Azure Data Lake Storage, Azure Data Share, Azure Databricks, Azure Stream Analytics, HDInsight, PowerBI Embedded, Azure Synapse Analytics, Data Factory, Event Hubs, R Server for HDInsight, Microsoft Graph Data Connect, Azure Purview, etc.
Compute	Build robust applications with high-level cloud compute capacity	API Apps, App Service, Azure Cycle Cloud, Azure Functions, Azure Kubernetes Service (AKS), Azure Quantum

Category	What are these categories?	Azure services
	and scalability features	Preview, Azure Spot Virtual Machines, Azure Spring Cloud, Azure VMware Solution, Azure Batch, Cloud Services, Linux Virtual Machines, Azure Container Instances, Azure Static Web Apps, VM Scale Sets, Azure Virtual Machines, Azure Virtual Desktop, Web Apps, Azure Dedicated Host, Azure VM Image Builder, etc.
Containers	Create, build, develop and manage containerized applications with modern integration tools	Azure Kubernetes Services (AKS), Azure Container Instances, Azure Container Registry, Azure Service Fabric, Web App for Containers
Databases	Fully managed and secure cloud database services	Azure SQL Database, Azure Cosmos DB, Azure Cache for Redis, Azure Database for PostgreSQL, Azure Database for MySQL, Apache Cassandra MI, SQL Server on Virtual Machines, Azure Database Migration Service, Table Storage, Azure API for FHIR, Azure SQL Database Edge, etc.

Category	What are these categories?	Azure services
Developer Tools	Services and development tools for engineers or developers working with cloud development and DevOps for Azure	Azure DevOps, Azure DevTest Labs, App Configuration, Azure SDKs, Azure Lab Services, Azure Pipelines, Visual Studio, Visual Studio Code
Integration	Services on Azure for different types of integration within Azure, hybrid or multi-cloud	Azure API Management (APIM), Azure Event Grid, Azure Service Bus, Azure Logic Apps, Azure Web PubSub Preview, Azure Healthcare APIs Preview
Networking	Connect cloud and on-premises infrastructure and resources using networking services	Application Gateway, Azure Bastion, Azure DNS, Azure Express Route, Azure Content Delivery Network, Load Balancer, Azure Front Door, Azure Firewall, Internet Analyzer, Azure Orbital, Private Link, VPN Gateway, Virtual WAN, Virtual Network, Traffic Manager
Internet of Things (IoT)	Create cloud solutions with IoT services	Azure IoT Hub, Azure IoT Central, Azure Sphere, Azure IoT Edge, Azure RTOS

Category	What are these categories?	Azure services
Identity + Security	Protect resources, data and identity on the cloud	Microsoft Entra ID, Microsoft Defender for Cloud, Azure Security Center, Azure Key Vault, Azure Sentinel, Information Protection, DDoS Protection, etc.

Table 2-2 describes some of the common Azure services. To view the updated list of services per category, [please visit the website](#). Other important Azure services will be covered in the other chapters of this book.

Overview of Azure Core Services

Microsoft consistently creates new solutions and continuously updates and improves Azure services.

Compute Services in Azure

Compute is the term used for computing resources. Compute services hosted in Azure provide computing resources like operating systems, networking, disks, processors, and memory. These compute resources are available quickly and on-demand for users. Every application is unique. An application can have many workloads that need more than one compute service.

As of this writing, Azure has more than 25 compute services available. Azure compute services enable us to build web and mobile applications, deploy and manage virtual machines, build apps in containers in the cloud, create batch jobs, and more. **Figure 2-2** shows an overview of compute services on the Azure Portal. Additionally, **Table 2-3** gives an overview of some of the common Azure compute services and their purpose.

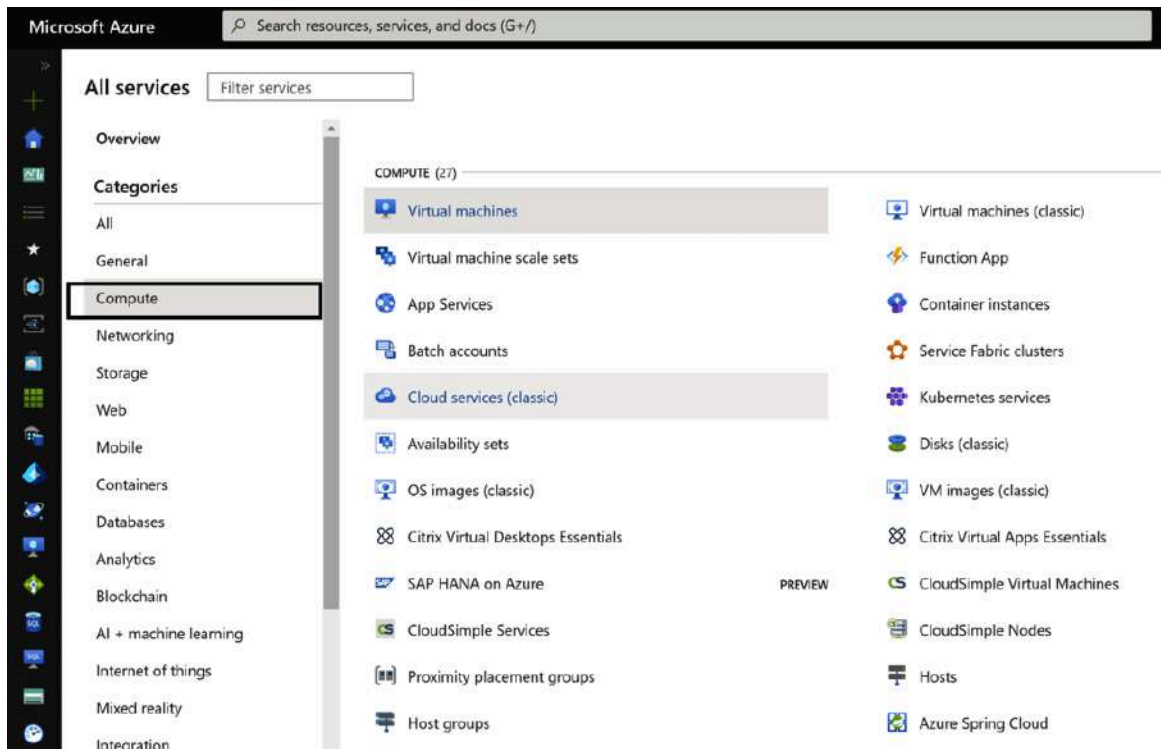


Figure 2-2. Overview of compute services in Microsoft Azure Portal

Table 2-3. Overview of Azure compute services

Azure compute service	What is it for?
Azure App Service	Build and develop web and mobile apps in a fully managed cloud environment
Azure Static Web Apps	Develop modern full stack web application quickly with Azure from a code repository
Azure Virtual Machines	Quick and manageable provisioning of virtual machines (Azure VMs) in different operating systems like Windows or Linux
Azure Virtual Machine Scale Sets	Create and provision multiple virtual machines (Azure VMs) with high availability advantage
Azure Spot Virtual Machines	Save money when you provision compute capacity you don't use for your workloads
Azure Functions	Develop serverless, event-driven applications, and stateful workflows
Azure Container Apps	Build and deploy fully managed modern apps and microservices using serverless containers
Azure Kubernetes Service(AKS)	Build managed Kubernetes containers on the cloud
Service Fabric	Build microservices and perform containers orchestration in different operating systems like

**Azure compute
service****What is it for?**

Windows and Linux

Technical details, use cases, and how you can get started developing with Azure compute services are covered in **Chapter 3**.

Networking Services in Azure

Networking services help secure both private and public cloud infrastructure. Users can customize their cloud networking setup and manage their network resources on demand.

Azure networking services allow Azure users to meet the demands of their infrastructure's workloads on-premises, using a hybrid approach, or in the cloud with high availability and enterprise-level **Microsoft Zero Trust-based security** on networking services. **Table 2-4** shows some of the common Azure networking services.

Table 2-4. Overview of Azure networking services

Azure Networking	What is it for?
Azure Virtual Network	Connect virtual machines using VPN connections
Azure Bastion	Secure and easy access to your virtual machines using private RDP and SSH that are fully managed
Azure Private Link	Access cloud Azure-hosted services with privacy
Azure Firewall	Protect your resources in the cloud with high availability and low maintenance firewall
Azure Load Balancer	Load balance your application connections and requests, both inbound and outbound
Azure ExpressRoute	Create private network connections between Azure data centers and on-premises infrastructure
Azure Traffic Manager	Route your network traffic for better performance
Azure VPN Gateway	Create secure private network connections in the cloud VPN

Learn more about the technical details and how you can get started with Azure networking in **Chapter 4**.

Core Azure Storage Services

The storage services in Azure offer great storage for any type of data objects, Azure Virtual Machine disk storage, reliable messaging storage, and other modern data types. They provide the benefits of high availability, durability, security, accessibility, and manageability. **Table 2-5** shows some of the common Azure storage services.

Table 2-5. Overview of Azure storage services

Azure Storage service	What is it for?
Azure Blobs	Store scalable binary data, text, or Data Lake Storage Gen2 big data analytics
Azure Files	Fully manageable file shares for deployments on-premises or for the cloud. Accessible anywhere through Server Message Block (SMB) protocol
Azure Queues	Using Queues you can collect large messages that you access via authenticated HTTP calls
Azure Managed Disks	Store block-level volumes for Azure Virtual Machines

Each core Azure storage service need to be integrated and associated with an Azure Storage account. A storage account in Azure is a container, as shown in **Figure 2-3**, of all Azure storage data like blobs, files, tables, and queues.

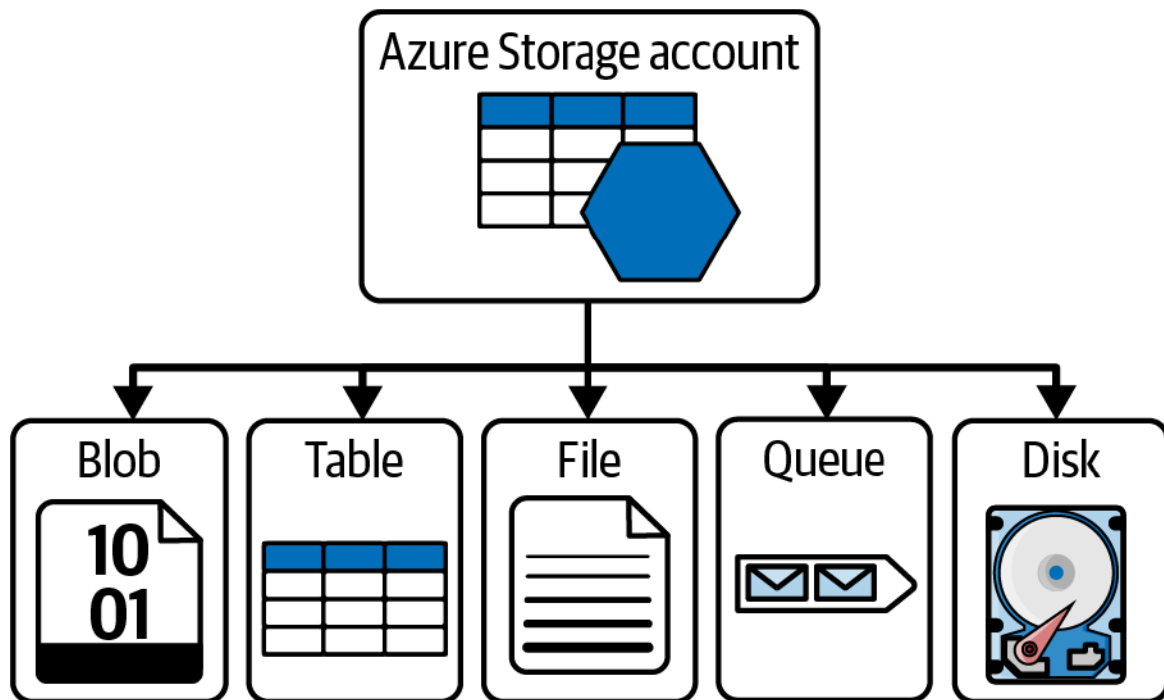


Figure 2-3. An Azure Storage account is needed for any of the core Azure Storage services

Core Azure Database Services

Azure has a great collection of database services to select from depending on the type of data structure you need for your organization: relational, NoSQL, in-memory databases, and other modern databases. [Table 2-6](#) shows the common Azure database services.

Table 2-6. Overview of Azure database services

Azure Database service	What is it for?
Azure SQL Database	Cloud-hosted SQL databases that are fully managed, intelligent, and secure
Azure Cosmos DB	Create and migrate NoSQL workloads to the cloud like Cassandra, MongoDB, and other NoSQL databases
Azure Cache for Redis DB	Build fast and scalable applications with Redis in-memory data store
Azure Database for PostgreSQL, MySQL, and MariaDB	Create fully managed and scalable databases for PostgreSQL, MySQL, and MariaDB
Azure SQL Edge	Build IoT edge-optimized SQL database engine with built-in AI

Learn more about the other storage and database solutions as well as further technical details in [Chapter 5](#).

Identity Management and Security Services

Secure your organization's cloud resources against threats using the identity management and security services in Azure. [Table 2-7](#) shows the common identity management and security services in Azure.

Table 2-7. Overview of Azure identity and security services

Azure Identity or Security service	What is it for?
Microsoft Entra ID	Secure users' identities and protect users of the entire organization using SSO and multi-factor authentication
Azure Information Protection	Protect your sensitive information on the cloud
Azure Key Vault	Allows you to control, manage, and secure your keys, connection strings, and secrets
Microsoft Defender for Cloud	Protect and detect threats for your workloads in Azure, on premises, and even in other cloud providers
Microsoft Sentinel	Gather intelligent security information and event management (SIEM) ^a and security orchestration automated response (SOAR) ^b solutions to protect your resources
Azure DDoS Protection	Protect applications in Azure from distributed denial of service (DDoS) attacks ^c

Azure Identity or Security service What is it for?

- a** Wikipedia, “Security information and event management,” <https://oreil.ly/VkCU8>
 - b** Rapid7.com, “Security Orchestration Automation and Response (SOAR) Tools and Solutions,” <https://oreil.ly/7XLQQ>
 - c** Cybersecurity & Infrastructure Security Agency, “Security Tip (ST04-015) - Understanding Denial-of-Service Attacks,” <https://oreil.ly/3GH4U>
-

NOTE

SIEM is the acronym for *Security information and event management*, a computer security system that can be used as a tool to collect, analyze, and perform security operations on computer systems, either applications or hardware. A SIEM system has features like collecting and logging data from resources within your environment, creating alerts for potential security anomalies, incident management, and data log visualization.

Learn more about the technical details and how you can get started with Azure user identity platform and security services in **Chapter 9**.

Developer Tools, Monitoring, and DevOps Services

Azure has different tools and services for cloud development, troubleshooting, monitoring, DevOps practices, infrastructure as code (IaC), and continuous integration/continuous delivery (CI/CD). **Table 2-8** shows the common Azure developer tools and DevOps services.

Table 2-8. Overview of Azure developer tools and DevOps services

Azure developer tools	What is it for?
Azure DevOps	All-in-one tool with great DevOps services for teams to collaborate, share code, track work, and deliver software projects
Azure DevTestLabs	Quickly create environments using reusable templates and artifacts
App Configuration	Store your application's configuration using scalable parameters
Visual Studio	Develop, debug, deploy, manage, and diagnose cloud-scale applications on Azure, using a full-featured IDE
Visual Studio Code	Write and debug code with a lightweight, fast code editor that runs on operating systems like Windows, Linux, and other supported operating systems

Other development tools for Azure such as Azure Developer CLI (azd), GitHub CLI, GitHub Copilot, and Microsoft Dev Box are discussed in [Chapter 14](#).

[Table 2-9](#) shows the common monitoring services in Azure.

Table 2-9. Overview of useful Azure monitoring tools

Azure monitoring tools	What is it for?
Azure Monitor	A great tool for maximizing application performance by collecting, analyzing, visualizing, and automating telemetry data in Azure and on-premises environments
Application Insights	Provides features useful for application performance management (APM) such as live monitoring and automatic detection of performance issues
Azure Advisor	Innovative cloud assistance in Azure that helps you improve your deployments by recommending useful and actionable solutions to secure resources, save costs, and improve performance
Log Analytics	Allows you to edit, run log queries, and analyze the data collected by Azure Monitor Logs

Core developer tools, supported programming languages for Azure, monitoring, troubleshooting steps, and DevOps services will be discussed in detail in **Chapter 11**.

Cloud Migration and Hybrid + Multi-Cloud Cloud Services

Azure has migration and hybrid solutions that help organizations in their cloud adoption and migration journey.

Table 2-10 shows the common Azure migration, hybrid, and multi-cloud services.

Table 2-10. Overview of Azure services for migration, multi-cloud, and hybrid cloud

Azure monitoring tools	What is it for?
Azure Database Migration Service	Get guidance and useful tools for migrating databases from different on-premises resources to Azure
Azure Data Box	Device solution for data transfer of large amounts of data to Azure and edge compute
Azure Migrate	All the guides and tools you need to migrate to Azure
Azure Arc	Combine and unify on-premises, hybrid, and multi-cloud infrastructure
Azure Stack	Build and run innovative hybrid apps across cloud boundaries

Learn more about the technical details and how you can get started with Azure migration and hybrid cloud solutions in **Chapter 13**.

These are the major categories of core services in Microsoft Azure. Each of them along with some of the categories listed in **Table 2-2** will be described in detail with examples in upcoming chapters.

Core Architecture and Resource Management Concepts in Microsoft Azure

An organization using Microsoft Azure will need their Azure administrator to properly set up the core structure in resource management. This is the first and top structure that is required before you can add cloud resources in the cloud platform. An organization with different departments with members in different roles would find this beneficial.

There are four levels for organizing your organization's resources in Microsoft Azure. As shown in **Figure 2-4**, from top to bottom, we have Azure management groups, Azure subscriptions, resource groups, and resources.

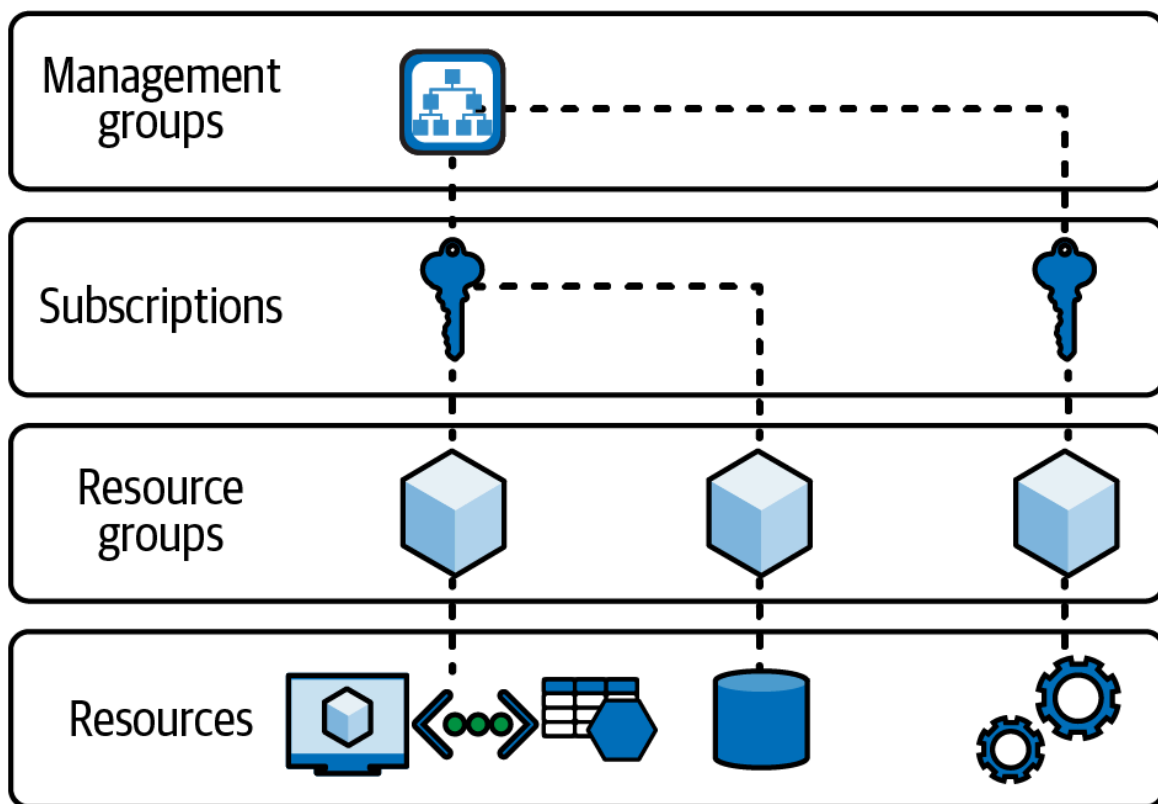


Figure 2-4. Four levels of Azure resources

Azure Management Groups

Azure management groups are the top level of the core structure for managing your cloud resources in Azure. The management groups are where Azure administrators manage everything about user access, compliance, and policies for subscriptions. The subscriptions within a management group automatically inherit the settings, conditions, and restrictions added in the group.

Azure *role-based access control* (RBAC) for all resources and role definitions is supported in the **root management groups**. A person in the organization with any Azure role can be assigned to the Azure management group. Those who have access and rights to the management group can group Azure subscriptions, see the organization's management groups and hierarchy, and, most of all, can control access to any Azure service or resource by creating and applying governance control and policies.

Azure Subscriptions

Subscriptions in Azure are like a big container for all user accounts and resources they have accessed or used within the subscription. Every subscription has a limit of resources that a certain user can create and use. As an organization, you can use subscriptions to control the monthly bill and resource costs in your organization or your own Azure account. Using Azure subscriptions, the organization can also control what resources the users create, update, or delete.

Azure Resource Groups

Azure users can group their services or resources using Azure resource groups. A resource group in Microsoft Azure acts as a logical container where resources like servers, web applications, databases, storage, monitoring, etc., are deployed, managed, and stored. Do not confuse a resource group with an availability set in Azure. The *availability set* is the logical group for virtual machines (VMs).

Azure Resources

The databases, servers, virtual machines, or web applications you create on the Azure platform are considered Azure resources. All resources or services you create must be added to a resource group, which acts as the logical container. In a resource group, you can have web apps, servers, monitoring, compute services, etc., in one place.

When you are creating an Azure resource as shown in **Figure 2-5**, you need to select an Azure subscription, a resource group, and the Azure region, for example, North Europe, where you want to add your resources.

[Home](#) > [Resource groups](#) >

Create a resource group ...

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#) ↗

Project details

Subscription * ⓘ

Resource group * ⓘ

Resource details

Region * ⓘ

Figure 2-5. Creating an Azure resource group in Azure Portal

You can also organize your Azure resources in categories by adding **tags** as shown in **Figure 2-6**. These resource tags are a key-value pair set composed of the name of your tag and the value. You can set tags for categorizing your resources for billing purposes. For example, you can apply tags for your resources for different environments like resources in Dev, Test, UAT, or Production.

Apply tags to your Azure resources to logically organize them by categories. A tag consists of a key (name) and a value. Tag names are case-insensitive and tag values are case-sensitive. [Learn more](#) 

Name ⓘ	Value ⓘ	Resource	
Environment	: Development	Resource group	
Billing	: Demo	Resource group	
	:	Resource group	

Figure 2-6. Using tags to categorize Azure resources

Azure Resource Manager

Azure Resource Manager (ARM) is also an important element in managing resources in Azure. ARM is the management and deployment service that provides users the capability to add, edit, and delete resources in Azure. By using ARM, the organization can manage user access control and organize resources securely even after deployment.

ARM templates are commonly used to automate deployments and to implement **infrastructure as code (IaC)**. Azure provides native support for IaC using the ARM templates and also **Azure Bicep** in the Azure Resource Manager.

Terraform is also supported on Azure. IaC creates a great advantage, and it enables deployment automation of the infrastructure in the cloud. Using infrastructure as code, you can automate your deployment by generating templates for the same environment every time. **The IaC** process minimizes the problems of environment drift during development releases.

Azure provides third-party support for other automated IaC platforms like **Terraform**, **Red Hat Ansible**, **Chef Automate**, and **Pulumi**.

Azure Bicep uses declarative, concise, and type-safe syntax to deploy Azure resources and is considered a domain-specific language (DSL). It promises the best developer experience when it comes to authoring IaC solutions in Azure. We will discuss the technical aspects in later chapters of this book.

Azure Geographies, Regions, Region Pairs, and Azure Availability Zones

Azure's global infrastructure is built to provide the best possible resiliency and high availability of cloud resources to its users. Azure infrastructure is composed of different geographies, regions, and availability zones in different countries across the world. Therefore, knowing where to provision your resources in Azure's global infrastructure will help an organization's cloud resource management, compliance, security, and speed.

Azure Geographies

As of this writing, Microsoft Azure has secured physical data centers worldwide in 140 countries. The **global infrastructure of Azure**, which also includes the infrastructure of Microsoft's cloud platform, is important in providing reliable, secure, and innovative smart cloud solutions. Learn more about the Azure data centers nearest to you by checking out **Azure geographies**.

Azure Regions

As you can see with the Azure geographies, there are data centers all over the globe in different regions and countries. Azure regions play a vital role in cloud computing for adaptability because each country or region has unique restrictions, policies, compliance, and rules. **Figure 2-7** shows an example of an Azure region with three availability zones.

Of all the leading cloud providers globally, Azure has the most global geographic regions. With the global and international market we have these days, being able to choose the location and geography of your resources offers flexibility and reliability. Choosing the right Azure region is also important when it comes to regulatory compliance and data policies such as the **General Data Protection Regulation (GDPR)** in Europe.

Before adding resources in Azure, your organization can check first if there are any legal standards or compliance issues in your geographic area.

Microsoft Azure has a **list of compliance offerings** for different sectors globally that you can review and learn more about.

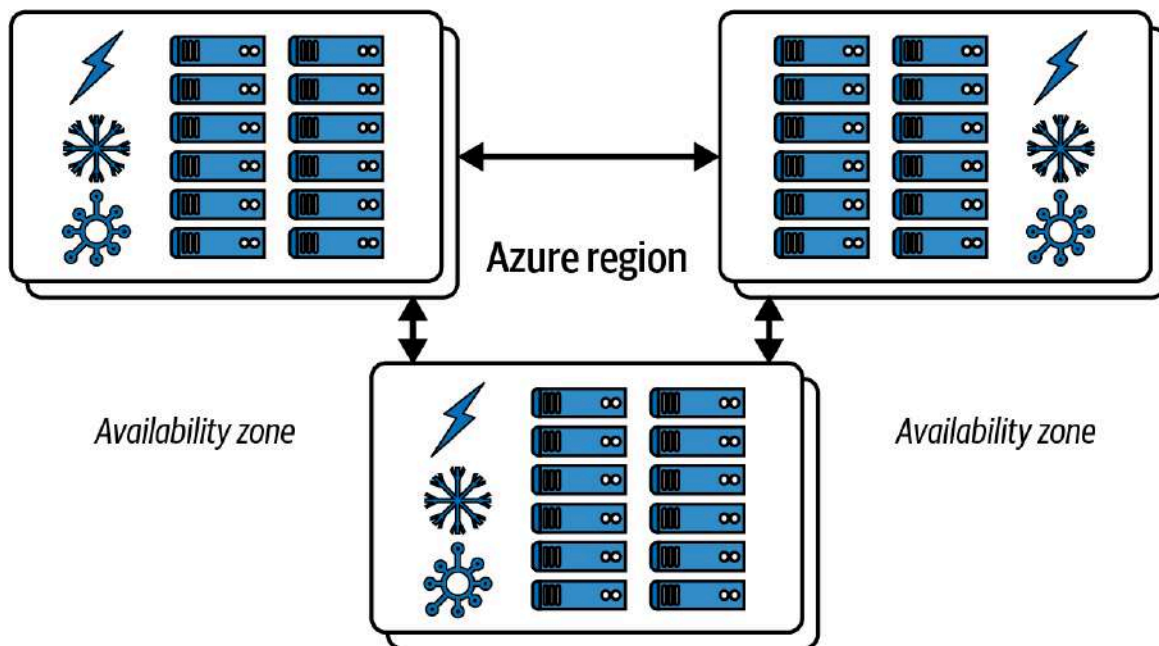


Figure 2-7. Example of an Azure region with three availability zones

Azure Region Pairs

Typically, availability zones are created by using one or more data centers. There is a minimum of three zones within a region. However, what if two of the zones go down because of a huge outage? It would be challenging to keep resources in operation if two data centers went out in the same Azure region. This is the reason users also have the option to use Azure region pairs.

Azure region pairs, as shown in **Figure 2-8**, involves pairing a region with another region within the same geography. By pairing regions within the same geography, users can replicate their resources like servers, virtual machines, databases, and storage in another location in case of power outages, natural disasters, etc.

If a region in a pair goes down, the services would failover automatically to the other region in its region pair.

An Azure region usually pairs with another region in the same geography. For example if the region is in Europe, then it is expected that its region pair is also located within Europe but at least 300 miles away. When possible, Azure prefers this minimum distance between physical data centers in a regional pair in case of natural disasters, widespread power outages, etc. The Azure regional pair should be within the same region for reliability in case of unexpected interruptions.

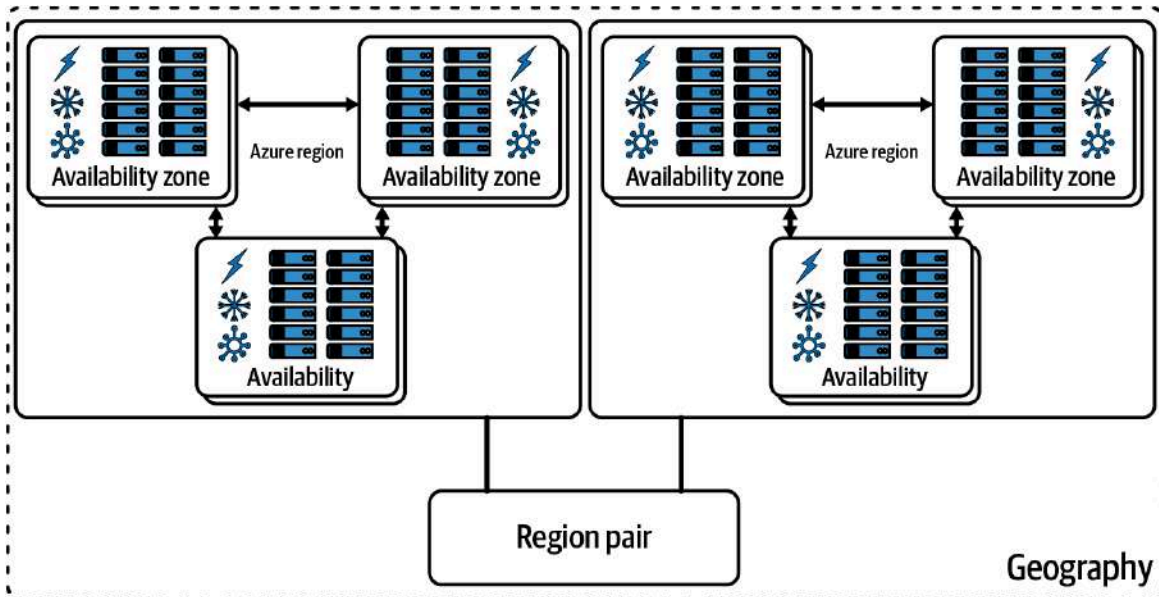


Figure 2-8. Azure region pairs

TIP

To learn about best practices, check out [Azure's business continuity and disaster recovery \(BCDR\)](#) for Azure paired regions.

Azure Availability Zones

If your organization wants to make sure that your applications and cloud resources in Azure are redundant in case of failure or problems, Azure availability zones are your solution. These zones are unique physical location or data centers within an Azure region that offer 99.99% SLA for virtual machine uptime.

It is important to note that availability zones are not available in all regions and Azure services. SQL Managed Identity, Azure App Services, and Azure Virtual Machines (Azure VMs) are supported.

Azure services supported by availability zones are meant to deliver resiliency, low latency, flexibility, and scalability. Learn more about the services that are supported by Azure availability zones by region by checking out [the Azure documentation](#).

An availability zone is composed of one or multiple data centers with independent facilities for power, networking, cooling, and support. The purpose of this isolation is to make sure that in case one of the zones stops working properly, the other zone continues to operate. These availability zones are connected and equipped with secure high-speed networks, which is important in running mission-critical resources for your computing, networking, storage, and data.

How are availability zones and [availability sets in virtual machines](#) alike and different? Availability zones are used to protect resources from complete system failures in an Azure data center while availability sets are used to protect applications from hardware failures within an Azure data center.

NOTE

If your organization has the requirement, you can replicate or transfer your data or resources into another availability zone for a cost in Azure. For example, [Azure VMs can be moved to an availability zone in a different region](#).

Cost Management in Microsoft Azure

Aside from organizing cloud resources using Azure management, resource groups, Azure tags, and making sure that your resources are in the right Azure region, you can also manage the financial aspects of your organization's cloud consumption.

Azure's cost management tools help organizations monitor their expenditures on cloud resources. This can be done by setting budget alerts and notifications for the appropriate billing or accounting team and also estimating possible monthly or yearly costs beforehand, which aids in cost planning and budgeting. These monitoring tools are available:

Azure Cost Management + Billing

An administrative section in Microsoft Azure where **billing and management of costs** can be controlled and monitored.

Total Cost of Ownership (TCO) Calculator

A tool that helps organizations estimate cost savings from **migrating workloads to Azure**.

Azure Pricing Calculator

A dedicated website where you can configure and **estimate the costs for Azure** products and features based on the use cases in your projects. You can save, export, and share these cost estimations, shown in **Figure 2-9**.

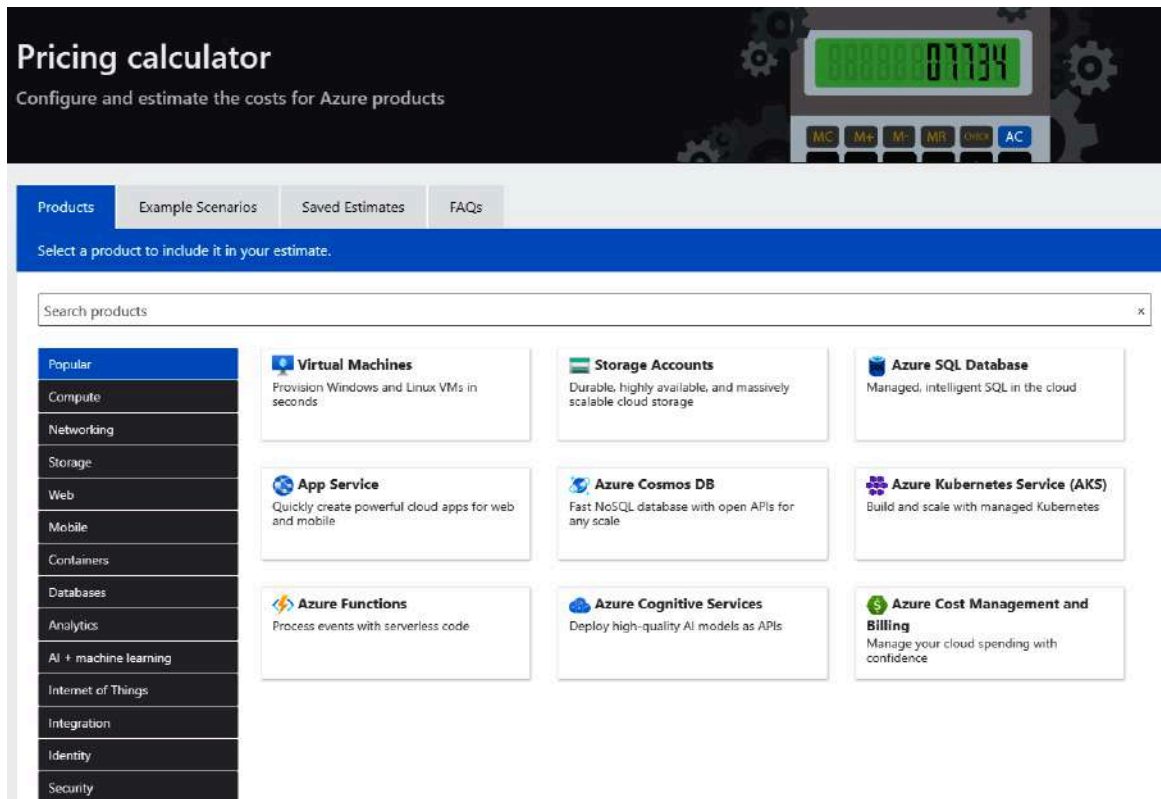


Figure 2-9. Estimated costs for Azure services or resources using Azure Pricing Calculator

TIP

Get an overview and learn some of the best practices in Azure billing and cost management by visiting the [Microsoft documentation](#).

User Identities, Roles, and Active Directories in Azure

In addition to structuring cloud resources using management groups, subscriptions and resource groups, it is also important to control user identities and access to these resources.

Microsoft Entra ID, as shown in [Figure 2-10](#), is the identity and user access management service of Microsoft. Users of Microsoft Entra ID can manage their identities, roles, logins, and access to internal resources as well as

permissions to external services like Azure Portal, Office 365, and other applications.

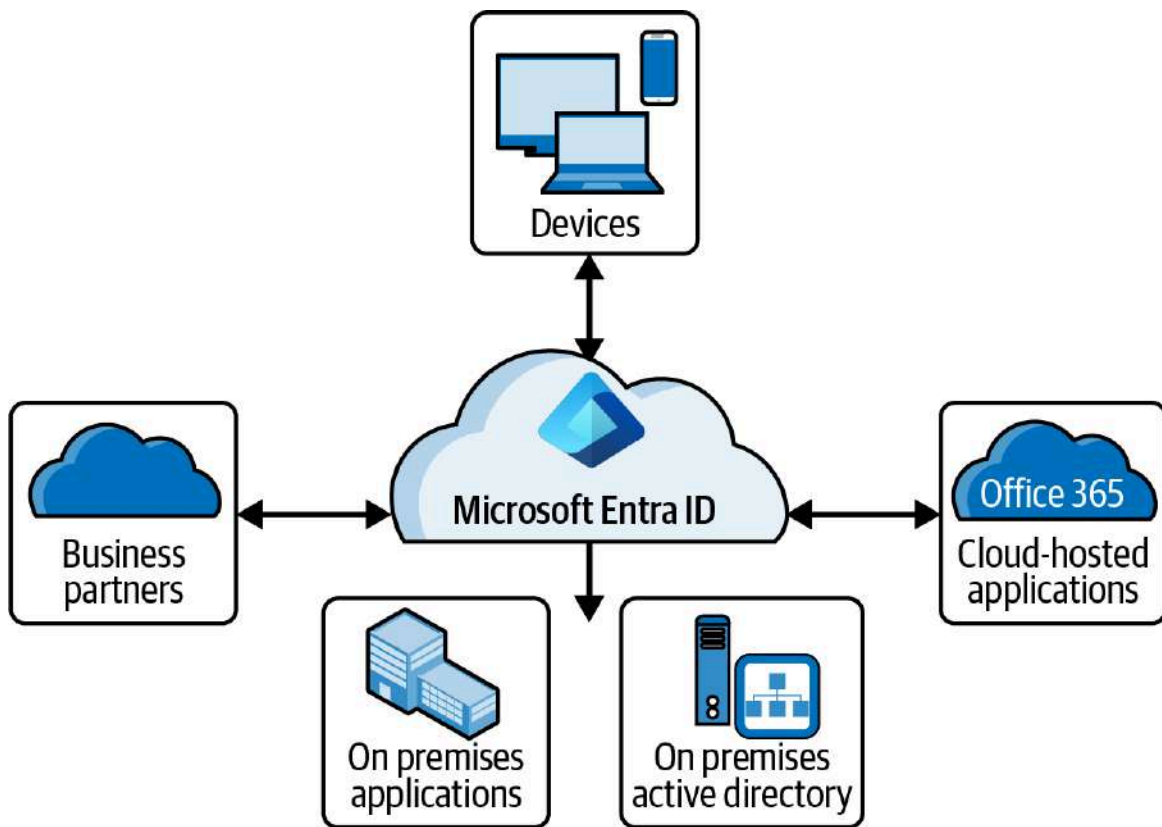


Figure 2-10. Microsoft Entra ID for user identity and access management

Azure Role-Based Access Control

It's important for any organization to be able to manage user access for cloud resources and resource groups in Microsoft Azure. Azure role-based access control (Azure RBAC) helps in authorization and user access management of resources in Azure. Management of identity using Azure RBAC helps in controlling what users can do and cannot do, depending on the role they have in the organization.

For example, you might allow database administrators to manage only Azure resources related to databases within a resource group. The access can be set up for developers or software engineers who work with application development on the platform.

To learn more about Azure RBAC, watch this [video about Azure RBAC](#).

Azure roles

To control access to resources in Azure, you need to set up and enforce user permissions by assigning *Azure roles*. A role assignment has elements such as a security principal, role definition, and scope (see [Figure 2-10](#)).

Security principal

A security principal is an object that represents a security identity that can be authenticated and authorized to access resources. Security principals are used in Azure roles to grant or deny resource permissions. They can be authenticated as a user, a security group, or a process that runs in a computer's account security context. A security principal or a security group can be identified in a computer's operating system using a unique **security identifier (SID)**. When you assign roles to a security principal, you're granting or denying permissions to access specific resources in Azure. For example, you could set the "contributor" role to a user principal, allowing users to create and manage resources in a specific Azure subscription. Understanding Azure roles is critical to ensuring the security and compliance of your Azure environment. You will learn more about security and identity management on Azure on [Chapter 9](#).

Role definition

A role definition sets permissions for Azure users or security principals to utilize Azure resources. Each role definition has a set of access controls, or actions, which helps determine which Azure resource activities are permissible. A role definition may allow reading, writing, and deleting Azure subscription resources. A security principal's role permissions apply to the assigned resources. The built-in role definitions, such as contributor and reader, address common resource access scenarios. Customize these built-in

role descriptions or create new ones to match company needs. You also have the option to customize role definitions, which lets you fine-tune resource access to meet your organization's needs. Azure role definitions allow flexible and granular resource access management.

Scope

The scope of a role determines the level at which the role assignment applies. It defines the set of resources the role assignment applies to and can be set at various levels in the Azure resource hierarchy, including the management group, subscription, resource group, and individual resource levels. You can minimize the risk of unauthorized and unknown access and the possible resulting consequences by specifying the scope of a job assignment. Managing Azure resource access and ensuring security and compliance requires understanding role assignment scope.

As noted throughout this section, the four levels of scope in Microsoft Azure are structured from top to bottom and follow the inheritance of a parent-child relationship.

Learn by Doing (Try It!)

Now that you have learned about the cloud services in Azure, this section presents some hands-on exercises that will help reinforce your understanding. For more hands-on exercises related to this chapter, see the [supplementary repository](#).

Microsoft Azure Portal Hands-On

1. If you already have an Microsoft Azure subscription, visit <https://portal.azure.com>.

2. If you don't have a subscription yet, register for a free Azure account on <https://azure.microsoft.com/en-us/free>.
3. Explore the Azure Portal's homepage, as shown in **Figure 2-11**, where you will see a left navigation bar with the default list of Azure services; the main page has the dashboard and search bar. On the upper right of the portal, you will be able to access the Cloud Shell, Directories + Subscriptions, Notifications, Settings, Help, and Manage Your Azure Account.

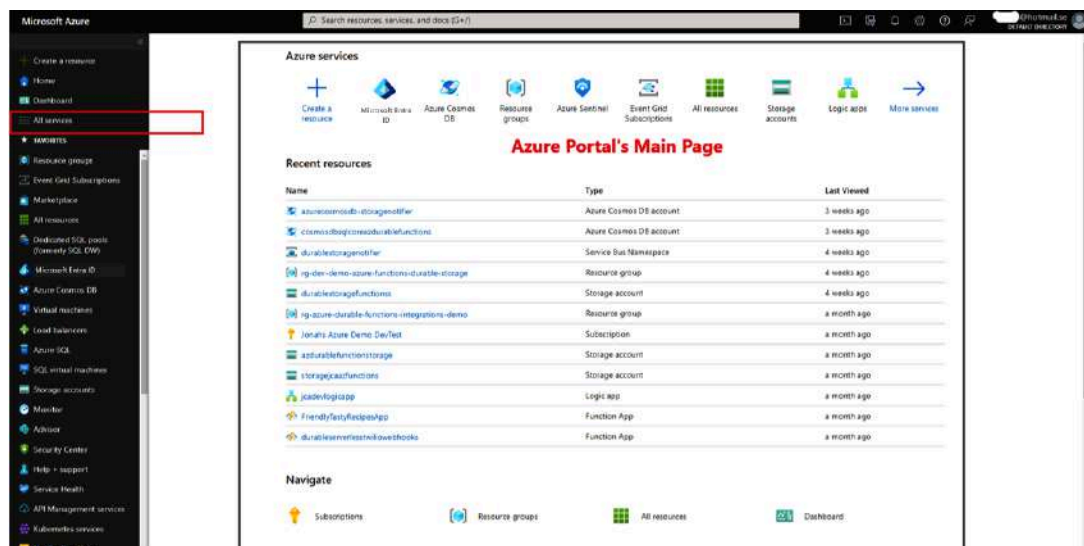


Figure 2-11. Microsoft Azure Portal when you have signed up for an Azure account and logged in

4. Click on *All Services* on the left panel and explore the different cloud services as shown in **Figure 2-12**.

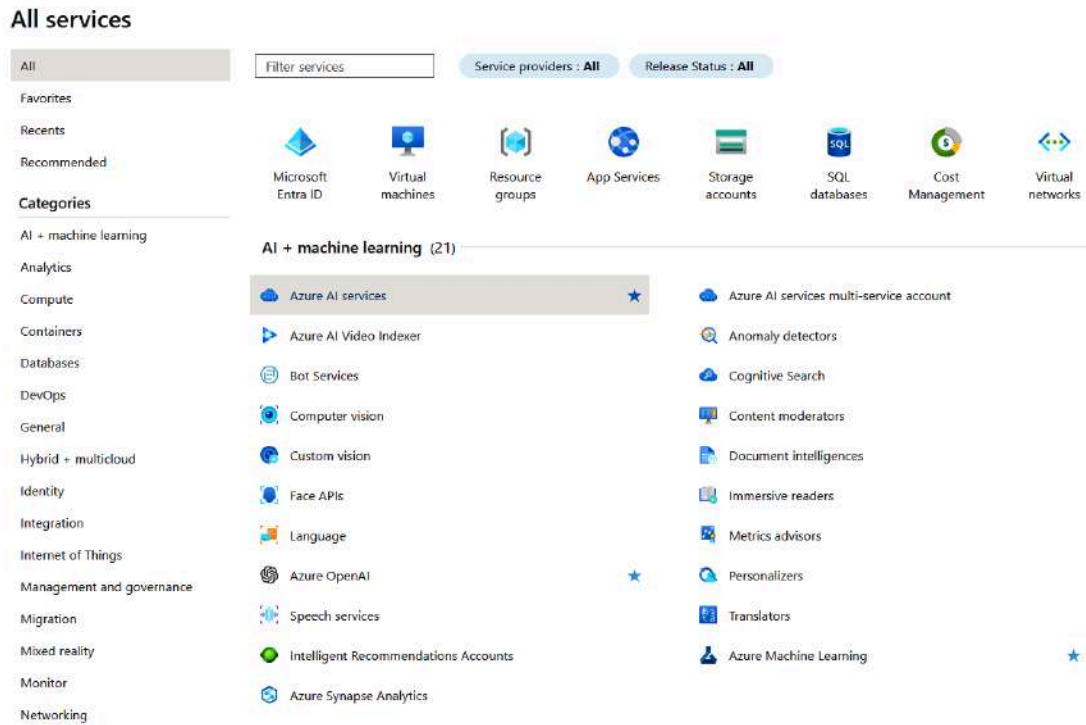


Figure 2-12. All services in Microsoft Azure Portal

5. Find free offerings in Microsoft Azure on the same page or visit <https://oreil.ly/I95fU>. Explore the free Azure services available to you for 12 months with your Azure free account.

Summary

In this chapter, you have learned the fundamentals of the public cloud provider, Microsoft Azure. You learned about the different Azure services in different categories, e.g., compute, storage, networking, databases, developer tools, integration, analytics, and more. Resources in Azure can be organized by and structured into different scopes: Azure management groups, subscriptions, and resource groups. Azure has helpful tools to manage costs and also to protect your data and resources in case of outages using availability zones and Azure region pairs. Microsoft Entra ID allows you to manage users and identities to control operation of your Azure resources.

Azure role-based access control (RBAC) is the user access control and identity management approach, and a role assignment is composed of three important elements: security principal, role definition, and scope. We also learned that Azure Resource Manager (ARM) templates and Terraform are used to automate deployments to Azure and to implement infrastructure as code (IaC).

Finally, the core Azure services mentioned in this chapter will be described in detail in the remaining chapters of this book.

Check Your Knowledge

1. Can you have two Azure region pairs in different geographies? (*True or False*)
2. Azure Marketplace is where Azure users can purchase and try out applications and services from other service providers and Microsoft partner companies. (*True or False*)
3. What are the top categories of core services in Azure?
4. Azure Virtual Machine Scale Sets is the compute service you need if you want to develop event-driven applications in a serverless environment. (*True or False*)
5. What is the difference between an *Azure region* and an *availability set*?

For the answers to these questions, see the **Appendix**.

Recommended Resources

“Azure Blob Storage Documentation.” Microsoft Learn,
<https://oreil.ly/U3HEO>.

“Azure Global Infrastructure.” Microsoft Azure documentation,
<https://oreil.ly/b5eJ3>.

“Azure Products.” Microsoft Azure documentation, <https://oreil.ly/Z8NjA>.
“Build in the Cloud with an Azure Free Account.” Microsoft Azure documentation, <https://oreil.ly/IoyT3>.

“Control and Organize Azure Resources with Azure Resource Manager.” Microsoft Learn, <https://oreil.ly/AHb3y>.

Kleppman, Martin. Designing Data-Intensive Applications. Sebastopol, CA: O’Reilly Media, 2017.

Microsoft Azure. “What Is Azure Role-Based Access Control (RBAC)? | One Dev Question: Arturo Lucatero.” YouTube video, June 23, 2020, <https://oreil.ly/7A8Ja>.

“Microsoft Azure Fundamentals: Describe Azure Architecture and Services.” Microsoft Learn, <https://oreil.ly/Ug5Zb>.

“Microsoft Azure Fundamentals: Describe Azure Management and Governance,” Microsoft Learn, <https://oreil.ly/tJ0aa>.

“Microsoft Azure Fundamentals: Describe Cloud Concepts.” Microsoft Learn, <https://oreil.ly/s1XTF>.

O’Connell, Ryan. “Azure Real World Hand-on Training for Beginners.” Udemy, <https://oreil.ly/VOjJM>.

Polkovnikov, Alexey. “Welcome to Azure Charts!” Azure Charts, <https://oreil.ly/oQ8ap>.

“Quickstart: Check Access for a User to a Single Azure Resource.” Microsoft Learn, July 18, 2023, <https://oreil.ly/9Fkjn>.

“What Is Azure?” Microsoft Azure documentation, <https://oreil.ly/KIhdc>.

¹ “Windows Azure: Microsoft’s Cloud Platform,” Microsoft Azure documentation, <https://docs.microsoft.com/en-us/archive/blogs/aniyer/windows-azure-microsofts-cloud-platform>

Part II. Compute, Networking, Storage, and Databases

This second part and its chapters focus on giving you an in-depth look at the different technologies in Azure for compute, networking, database, and storage categories, which are important in developing applications and cloud solutions. You also get an introductory overview of Azure Space and Azure Orbital, which are Azure's ground station as a service.

Chapter 3. Microsoft Azure Cloud Compute Services

Compute services, networking, and storage are the essentials of any public cloud. Azure compute helps customers build and host application workloads of various scales and natures. It works with your current development stack or helps migrate it to containers or serverless and adopts the best development practices.

—Alexey Polkovnikov, Sr. Cloud Solution Architect at
Microsoft and Founder of [Azure Charts](#)

Previously, we learned about Microsoft Azure as a cloud provider and reviewed Azure’s different core services, which gave us a better understanding of each cloud service.

In this chapter, we dig deeper into one core services category in Microsoft’s cloud platform, Azure compute, which helps you and your organization build cloud solutions using compute services, which will be utilized in existing development projects regardless of whether your applications are for migration, containerization, or modernization.

Azure Compute for Developing Fully Managed Systems

Azure provides computing services on-demand globally and with high availability. Azure compute is the category name for cloud-based computing services in Azure, and it offers on-demand computing resources such as high-capacity virtual disks, memory, networking, processors, and robust virtual operating systems. They are available on a [pay-as-you-go](#) basis or with other pricing models that can be set up and created quickly. The ability to provision compute resources with pricing and scaling

flexibility can be challenging if you are provisioning compute resources on premises using traditional IT infrastructure.

You can find the different compute services in Azure Portal by navigating to All Services and then selecting the Compute category as shown in **Figure 3-1**.

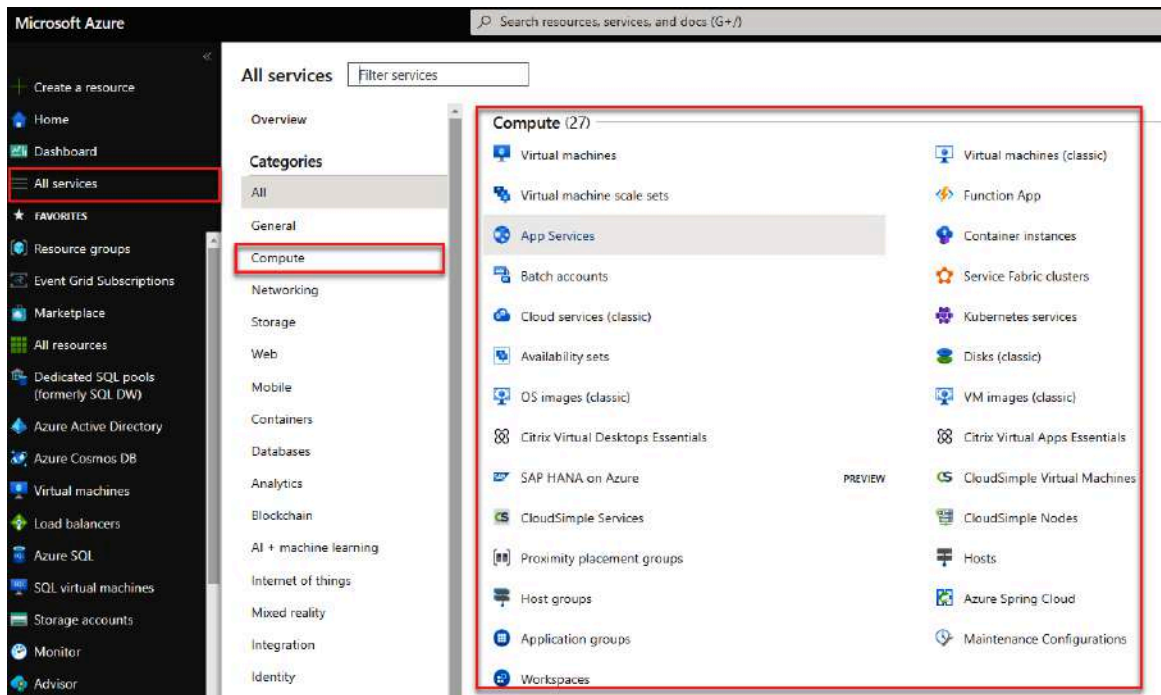


Figure 3-1. Azure compute services

NOTE

Azure compute services are used for running and hosting the workloads for our applications. Some familiar Azure compute resources are Azure Virtual Machines (VMs), Azure VM Scale Sets, Azure App Services, Azure Container Apps, Azure Functions, and Azure Kubernetes (AKS). These compute resources are categorized in different deployment models, such as infrastructure as a service (IaaS) or platform as a service (PaaS), and used in combination with serverless or separately depending on the type of applications and business requirements that need to be solved.

The computing technologies shown in **Figure 3-2** will be useful in hosting your applications in the cloud while gaining the benefits of having resources and workloads fully managed, with better security and the

opportunity for rapid development and innovation. Systems that are still on premises can use Azure compute services to improve their systems and save time and money compared to the traditional IT infrastructure.

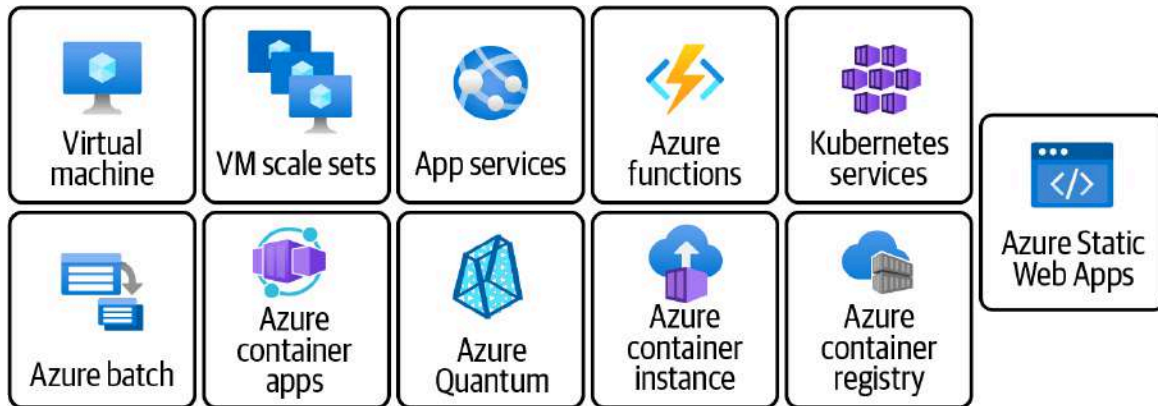


Figure 3-2. Common Azure compute services

Azure Virtual Machines and Virtual Machine Scale Sets

In addition to the commonly used compute services, another way to deploy applications and databases to Azure is using Azure Virtual Machines (Azure VMs) and Azure VM Scale Sets, which provide opportunities for hybrid deployment both on premises and in the cloud. This enables you to deploy your applications and databases to Azure in a more cost-effective and scalable way than traditional IT infrastructure.

These services enable us to quickly create virtual machines on the cloud on demand anytime and anywhere, especially for those who prefer implementing IaaS solutions for their business needs.

For example, Azure VM Scale Sets allows you to create thousands of Azure VMs on the cloud in a few minutes with autoscaling and load-balancing features that will help you meet the demands of your workloads. Azure VM Scale Sets can also be used as Azure DevOps Agents for CI/CD pipelines.

Azure VMs are compatible with operating systems on different platforms like Windows, Linux, and others. Azure VMs provides high availability,

scalability, reliability, and fully managed VM features for users and administrators. They are also commonly used by those seeking lift-and-shift cloud migration scenarios. The Azure VMs can also be used as agent pools for deployments on Azure DevOps to Microsoft Azure.

Lift-and-shift cloud migration has critical benefits such as minimal code refactoring and faster migration. This cloud migration strategy is ideal for organizations that want to move their existing on-premises IT infrastructure to the cloud without making significant changes to the existing architecture, source code, and processes. By utilizing IaaS solutions and even container solutions for cloud migration, organizations can save time, money, and resources by restructuring and modernizing the applications.

Cloud migration options and services in Azure will be discussed in greater detail in [Chapter 13](#).

As you may recall from [Chapter 1](#), virtualization relates to the evolution of cloud computing, and it is what drives the IaaS solutions we have today.¹

Azure Virtual Machines

As mentioned, Azure VMs are an IaaS cloud service model, which means that you have complete control of the infrastructure plus the advantage of scalable computing resources. Having complete control of the infrastructure also means that you are responsible for maintaining it.

Azure VMs may be ideal if your organization wants full capabilities in installing and configuring your infrastructure without spending money buying physical hardware or expanding data centers. Additionally, Azure has options to allow Azure VM users to manage security, monitoring, updates, and patches to the operating systems.

Through virtualization in the cloud, you can deploy and host your applications on your desired OS on Azure VMs in different servers and with shared storage, as shown in [Figure 3-3](#).

Development and applications

Through its cloud-based services, Azure VMs provide the flexibility of virtualization, and it also gives you control over how you want to host your applications. A wide range of computing solutions exists for development, testing, running applications, and extending your data center. It gives some users the option of using custom and open source software systems with user-defined configurations. This gives the flexibility and benefit of faster application deployment (seconds instead of weeks).

TIP

[Azure Lab Services](#) and [Azure DevTest Labs](#) are useful if you want to set up different development, test, and lab environments for your team or for learning purposes.

Making use of Azure VMs as part of an IaaS solution is useful, especially if you want to take control of your on-premises resources or infrastructure. However, there are some important and practical things to consider before provisioning Azure VMs for your organization on the cloud, as described in [Table 3-1](#).

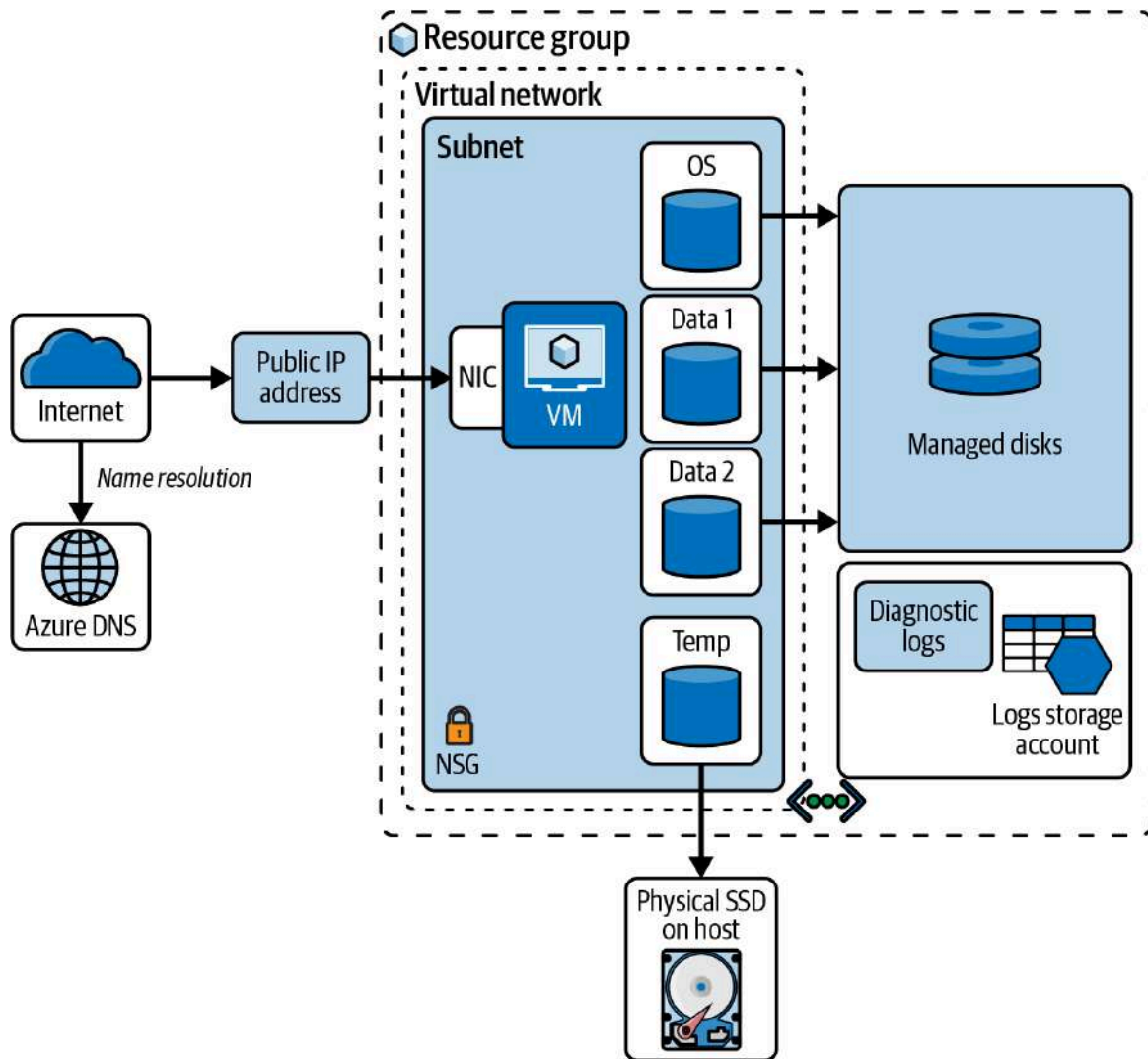


Figure 3-3. Overview of Azure Virtual Machines

Table 3-1. Things to consider before creating Azure VMs

Consideration	Description
Basics of networking	Azure uses virtual networks (VNets) to secure connections for Azure VMs and other cloud services in the platform. Any Azure VNet Resources that belong to the same private network can be accessed.
Name of the VM	Choosing an appropriate and consistent name for your Azure VM is part of the provisioning process. The VM name is configured as part of the operating system and is also used to define and manage it as a resource.
VM size	Options to choose VM sizes are available so that you can choose the right combination of memory, compute, and storage for type of workload needed.
Knowledge of VM pricing models	Being aware of the pricing models for Azure VMs is important to prevent unwanted costs. Azure VM pricing models come in five different types. Each option will vary not only in price but also in performance and availability.
VM storage	Azure VMs have at least two virtual hard disks (VHDs) used for storage of the operating system (OS) and temporary storage. Determining the appropriate VM storage prior to provisioning is important so that the VMs you create are provisioned to handle the workloads you need. Check out Microsoft's recommendation for virtual machine sizing guidelines .

Consideration	Description
Choosing the right operating system	There is a variety of OS images you can install into the Azure VM, including several versions of Windows and different types of Linux. Choosing the appropriate OS prior to VM provisioning is vital. Usually the choice of OS will affect the cost of hourly compute pricing as Azure combines the cost of the OS license into its price. Using Azure Cloud Shell , you can easily see the list of the latest Azure VM images and OS versions using the command <code>az vm image list</code> and other required parameters.

For more details and information about Azure VM provisioning, visit [Microsoft's checklist for creating an Azure Virtual Machine](#) and evaluate the different pricing models for [Linux VMs](#) and [Windows VMs](#).

Azure Virtual Machine Scale Sets

Azure Virtual Machine Scale Sets is a compute service that allows you to deploy, run, and manage a scalable set of multiple VMs. With all VMs configured the same, Azure VM Scale Sets are designed to support autoscaling. There is no requirement for pre-provisioning of VMs, which makes it easier to build services at a large scale. This can be useful in developing solutions with big data, big computing, and even workloads that are containerized.

When the demand for workloads increases, there is an option to add extra instances of virtual machines. When the need for workload decreases, VM instances can be deleted or scaled down. The scaling options and processes for Azure VM Scale Sets can be done automatically or manually, depending on your workload needs. It is also possible to dynamically combine autoscaling and manual scaling if necessary. You can set autoscaling

options for vertical or horizontal scaling on Azure VM Scale Sets, as shown in [Figure 3-4](#).

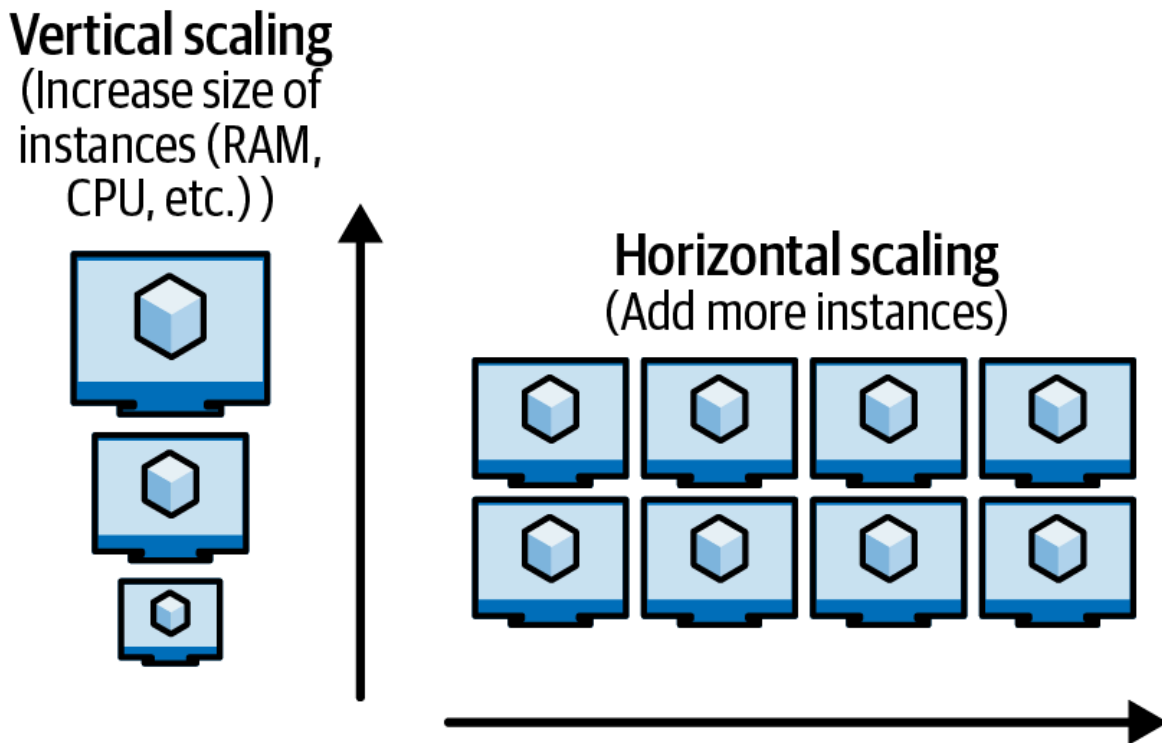


Figure 3-4. A simple illustration of vertical and horizontal scaling

Scaling Options for Azure VM Scale Sets

Azure VM Scale Sets are meant to avoid the high costs of multiple virtual machines in the cloud. Therefore, new instances of virtual machines are recommended for provisioning when necessary. Just like the standard Azure VMs, you have the option stop or deallocate to save money for running the compute, but you will still be charged for the storage configured for that virtual machine.

Azure's **autoscaling** offers the option to scale up and down the number of compute resources being allocated based on demand at a certain time. You can scale your virtual machines using horizontal scaling or vertical scaling.

A good example of typical horizontal scaling is when you add or remove some VMs from a scale set. On the other hand, with vertical scaling, you

are adding additional resources to your VMs, like upgrading the memory, disk memory, and CPU power.

Finally, both Azure VMs and Azure VM Scale Sets are great options for hosting databases, storage, and applications for your organization. They are both ideal, especially if your organization is still in the process of getting started in your cloud migration journey and prefers to have some of its resources, data, and workloads on premises. In **Chapter 13**, you will learn more about cloud migration and building solutions on Azure for hybrid and multicloud use cases.

AZURE HYBRID BENEFIT AND AZURE SPOT VIRTUAL MACHINES

As your workloads and VMs increase in the IaaS model, using Azure VMs can be expensive. Azure has optional services like **Azure Hybrid Benefit** and **Azure Spot VMs**. Azure Hybrid Benefit is a licensed benefit that helps organizations save money by reducing the costs of running workloads in the cloud. The benefit gives the option for organizations to use their own on-premises Windows Server and SQL Server licenses that are Software Assurance-enabled on Azure.

The **Azure Hybrid Benefit Savings Calculator** is a useful tool for analyzing the financial benefits you will gain. Learn more about the benefits of this licensing and how it will save money on Azure VMs by using your organization's on-premises licenses for Windows Server and SQL Server.

In addition, **Azure Spot VMs** can help reduce costs through reducing unused capacity.

Azure App Service

Azure App Service is considered a platform as a service (PaaS) and is an HTTP-based service that enables you to host web applications on the cloud. If you do not need full control of your infrastructure, like the demand for using Azure VMs, then App Service is one of the most common fully managed PaaS services for web application development.

It allows developers to easily create, deploy, and manage cloud-based web apps and APIs. With App Service, developers can concentrate on writing

application code rather than fretting about the infrastructure that underlies the application.

App Service has the following applications and features:

Built-in integration capabilities

App Service offers built-in integration with central source control systems such as GitHub and Azure DevOps to facilitate deployment. Developers have various application deployment options, including code deployment, Docker containers, and Azure Functions integration. The feature of flexible integration with other Azure services such as Azure SQL Database, Azure Blob Storage, and Azure Functions helps developers create a comprehensive solution that is scalable, dependable, and secure.

Support for multiple languages and frameworks

This service supports multiple programming languages and frameworks, such as NET, Node.js, Java, PHP, and Python. The support for various programming languages and frameworks allows developers to select the language and framework that most closely fits their project's needs.

Autoscaling

App Service can autonomously scale up or down application instances based on traffic load. The autoscaling features help ensure that the application's efficacy is maximized while costs are minimized.

High availability and fault tolerance

App Service also offers high availability and tolerance capabilities for failures, including geo-redundancy, automatic failover, and backup and restore capabilities. This

guarantees that the app is always available to users and can recover rapidly from failures.

Hybrid Connections

Azure App Service provides a feature called Hybrid Connections, which allows developers to securely connect their App Service applications to on-premises resources over the internet without requiring complex configuration or infrastructure changes. The Hybrid Connections feature creates a secure outbound connection from the App Service to a relay agent hosted in the Azure Relay service. The relay agent acts as a bridge between the App Service and the on-premises resource, allowing the App Service to communicate with it securely. To use this hybrid integration feature, the developer must set up a relay agent in Azure and configure it with the appropriate settings. Once it is set up, it can connect the App Service to an on-premises resource like a database, web service, or TCP endpoint by configuring the connection string to use the Hybrid Connection endpoint. This feature of Azure App Service is handy for enterprises with legacy systems or on-premises resources that cannot be easily moved to the cloud. It allows them to leverage the benefits of Azure App Service while still being able to access their existing systems and data. Additionally, it eliminates the need for complex VPN configurations and firewall rules, making it easy to use and manage.

Overall, App Service is a powerful PaaS solution that offers a variety of features and capabilities for developing and deploying web-based applications and APIs on Azure.

Figure 3-5 shows an example architecture from the Microsoft documentation website with implementation of an App Service in an App

Service plan. The web app has integrations with other services like Microsoft Entra ID for authentication, Azure Front Door, Azure Queue Storage for messaging, and Azure Functions for serverless backend logic to database storage for Azure SQL and Cosmos DB databases using Redis Cache. The application has its Azure SQL Databases on SQL Server and integrates with Azure Monitor for logging the metrics and diagnostics for the web application. The Azure Key Vault is used for safely storing secrets, keys, and connection strings for the applications. Deployment slots in the Azure App Service is used to help with zero-downtime deployments between different versions of the application to production.

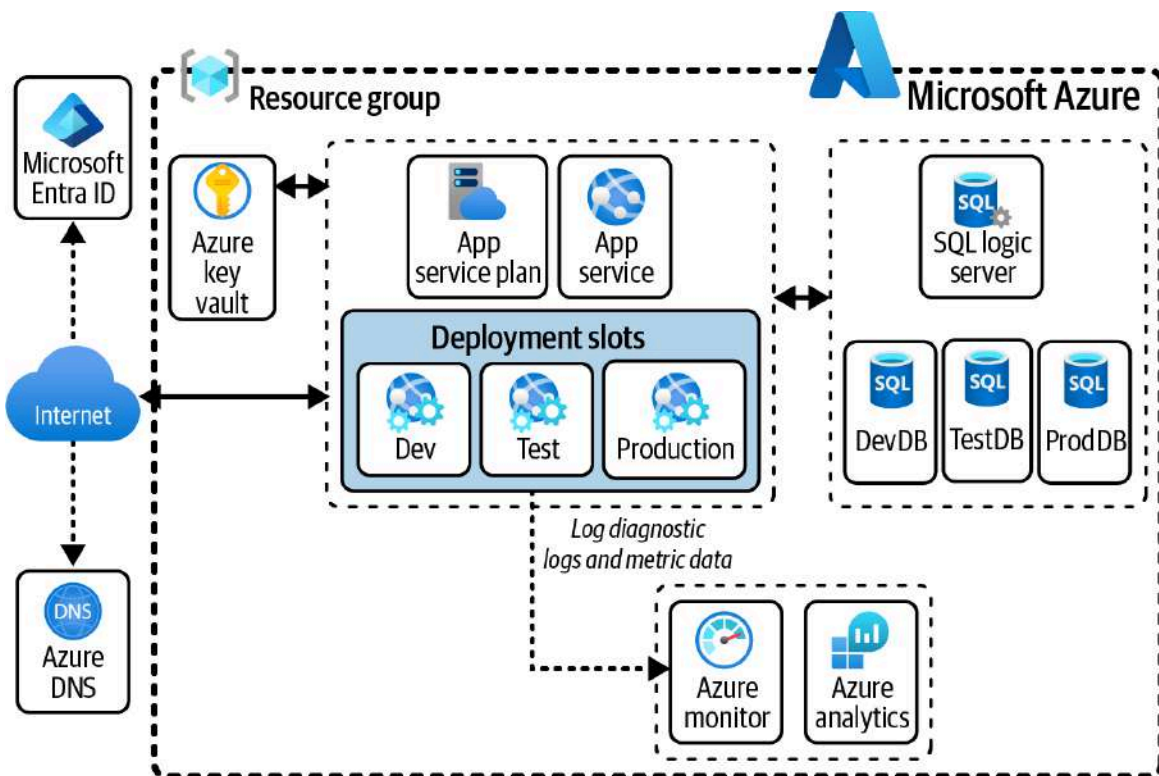


Figure 3-5. A basic example of a web application implementing Azure App Service in different environments (Dev, Test, and Production) using deployment slots

Modern application development is more efficient using App Service because it has web applications for autoscaling, automation, security, workload balancing, monitoring, and support for continuous delivery and continuous integration (CI/CD) with Azure DevOps, Docker Hub, GitHub, etc. A list of the key benefits of using Azure App Service is in [Table 3-2](#).

The **Azure App Service plan** determines the type of pricing model and how much you pay for the Azure compute resources consumed monthly.

Table 3-2. Benefits of using Azure App Service

Benefits	Description
Multiple languages and frameworks	App Service has first-class support for different frameworks and programming languages including just.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run PowerShell and other scripts or executables as background services.
Managed environments	App Service automatically patches and maintains the OS and language frameworks, which enable developers to spend more time writing code and worry less about setting up the platform.
Containerization	Containerize or dockerize your application and host it on a custom container (Linux or Windows) in App Service. Using Docker Compose, you can deploy and run multi-container apps.
Optimized DevOps	Set up continuous integration and deployment with Azure DevOps, GitHub, BitBucket, Docker Hub, or Azure Container Registry. Promote updates through test and staging environments. Manage apps in App Service using Azure PowerShell or the cross-platform command-line interface (CLI).
Global scale and high availability	Host apps anywhere in Microsoft's global data center infrastructure with autoscaling or manual scaling capabilities, and high availability.

Benefits	Description
Integration with on-premises and SaaS platforms	A wide variety of connectors are available for enterprise systems, SaaS services, and external services. Data from on-premises systems can also be accessed using features like Hybrid Connections and Azure Virtual Networks.
Security and compliance	You can secure your App Service by adding Microsoft Entra ID for user authentication or Single Sign On (SSO) with external services like Google, Facebook, Twitter, or a Microsoft account. You can set-up Managed Service Identities and restrict some IP addresses for security.
Application templates	If you need to use app templates, there are a list of them in the Azure Marketplace , such as WordPress, Joomla, and Drupal.
Integration with Visual Studio and Visual Studio Code	Development teams can code, deploy, and debug using the great IDEs like Visual Studio and Visual Studio Code.
API and mobile features	App Service provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more
Serverless logic	Web apps can be integrated well with serverless code as used in Azure Functions. This means that you can run serverless code and scripts and call API endpoints on-demand without having to

Benefits	Description
	manage infrastructure or explicitly provision. Using the consumption plan pricing model, you only pay for the compute time when your Azure Functions instance runs.

Azure App Service allows us to develop and host applications and APIs using our preferred programming language as PaaS without managing infrastructure. App Service also supports **single-page application (SPA)**, open source content management systems (CMS) like WordPress, Apps on Azure Arc, Azure Static Web Apps for static HTML sites, multi-container apps using Docker, etc.

Azure Web App for Containers

Azure Web App for Containers enables you to deploy your applications as containers in the cloud with the capability to use custom Docker images with support for autoscaling and easy deployment. Container registries like Docker Hub or Azure Container Registry can be used to pull the images of your application. An example architecture for Azure Web App for Containers is shown in **Figure 3-6**. The illustration describes how Web Apps for Containers for Azure App Service can be used in migrating applications hosted on containers. This implementation is possible with integration with other Azure services such as the Azure Container Instance, Container Registry, Application Insights, and other Azure database services.

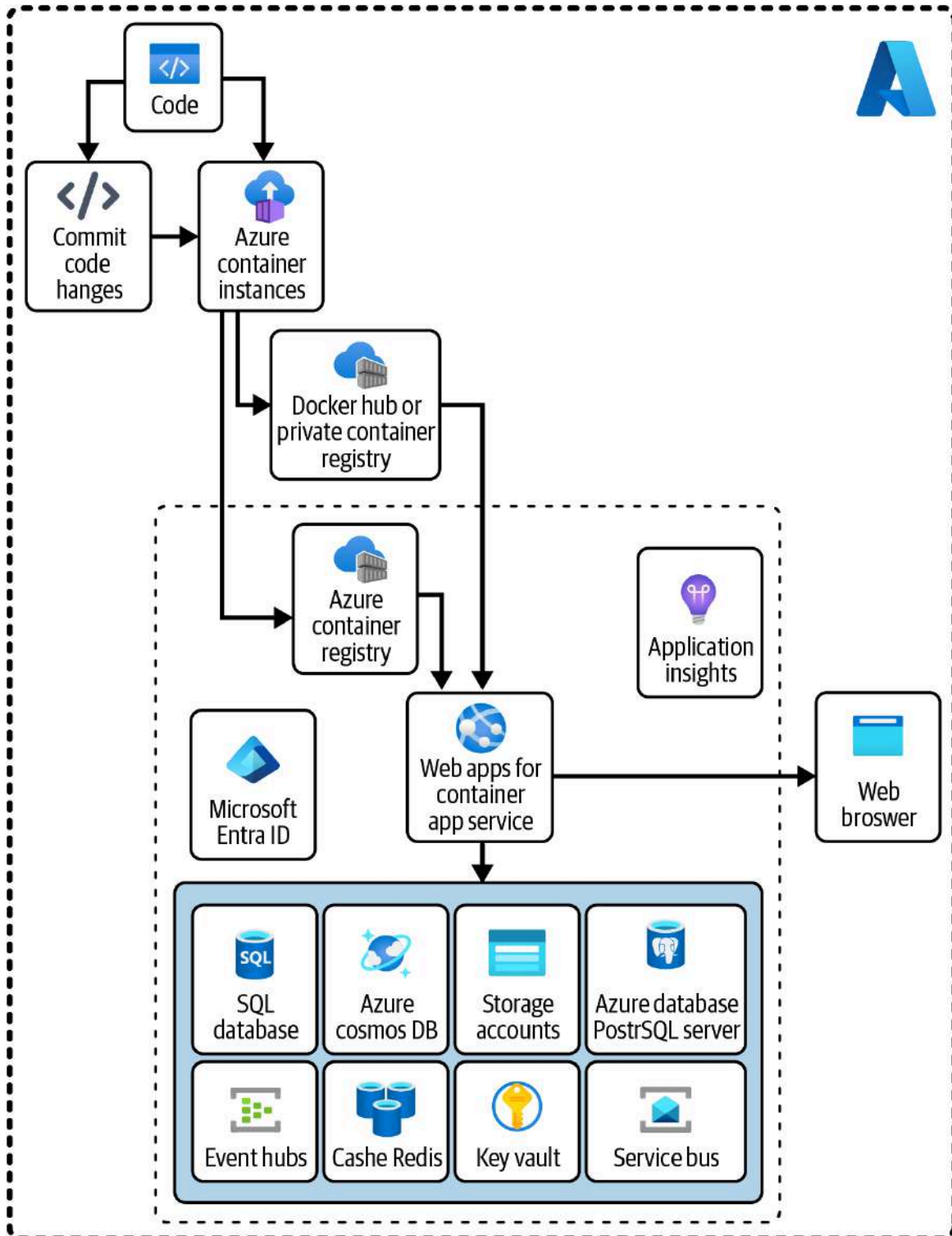


Figure 3-6. Example architecture of Azure Web App for Containers

Azure Web App for Container also can develop a **custom container** and **multi-container (Preview)** apps using **Cloud Shell** and **Docker**.

Static Web Apps

Azure Static Web Apps is a single-page application (SPA) service that allows you to develop full stack SPA web applications that complete CI/CD integration with a code repository platform like GitHub or Azure DevOps. Static Web Apps are reliable, fast applications you can configure using the Always On feature. The Always On configuration keeps your application loaded in memory, an advantage if event-driven WebJobs need to run in the background.

Prerequisites for developing Static Web Apps are an Azure and GitHub account as the source code needs to be set up with [GitHub Actions](#) for the automation of the deployment workflow.

Static Web Apps in Azure give us the option to build modern web applications with modern frameworks and libraries. For example, this Azure service can be integrated with [Blazor Web Assembly](#) applications, React, Angular, Vue, or Svelte. It is also designed to integrate well with serverless development by using Azure Functions as a backend or for APIs. If you need guidance in choosing between the traditional web applications compared to using SPAs, check out [Choose Between Traditional Web Apps and Single Page Apps \(SPAs\)](#).

Key Benefits and Uses of Azure Static Web Apps

If you think your application must expose a rich user interface with many features, Static Web Apps doesn't require reloading the page as users take actions or navigate between areas of the app.

The use cases for and benefits to using Static Web Apps are:

- Build modern web applications including WebAssembly applications with Blazor
- Bring your own functions support for serverless development especially if you already have existing Azure Functions to implement

- Seamless developer experience and CI/CD with GitHub Actions or Azure DevOps
- Globally distributed static content closer to the location of the users of your web applications
- Free and autorenewable SSL certificates for both free and standard pricing models
- Custom domain support with TLS
- Seamless security with support of reverse-proxy when calling APIs, which requires no CORS configuration²
- Authentication with common identity providers such as Microsoft Entra ID, GitHub, Twitter, etc.

PRICING PLANS FOR AZURE STATIC WEB APPS

There are free and standard pricing models for Static Web Apps. Learn more about the list of features of each plan in the [Azure Static Web Apps documentation](#) and be sure to reference the [quotas](#) page before determining the best fit for your needs.

Oryx is the build engine of Azure Static Web Apps. It helps in building the frontend and the API in your web application. It is a smart build engine because it automatically builds the steps and executes them for you depending on the type of framework you have implemented in the app.

WARNING

Since Azure Static Web Apps dynamically allow us to build SPAs using the frontend framework of our choice, configuration differences exist. The build configuration values needs to be specific to the type of framework or library used for the frontend.

Learn more about it in [“Configure Front-End Frameworks and Libraries with Azure Static Web Apps”](#).

In conclusion, Azure Static Web Apps are ideal for modern SPAs and for hosting static web content with support for global distribution, scalability, speed, security, integration, and ease of use for both development and deployment.

Serverless Compute Services

A powerful and promising compute service in Azure is the serverless compute service. With the technology of **serverless computing** developers can build applications without maintaining and worrying about the underlying infrastructure. Technically, the servers and infrastructure of applications in serverless compute are automatically provisioned, which means they are managed and scaled by Azure, the cloud service provider. As a developer, this helps you focus on writing the code and application development.

Benefits of Serverless Architecture in the Cloud

In simple terms, serverless computing is a model where backend services are provided as a cloud service.

For example, an organization can use the serverless or backend services from Azure with the advantage of having a consumption pricing model, which means you pay as you use. **Figure 3-7** illustrates how serverless computing can help save money compared to building and maintaining servers and infrastructure the traditional way.

On-premises IT infrastructure is expensive because it relies on our own physical hardware and data centers to host systems and web applications on our own servers.

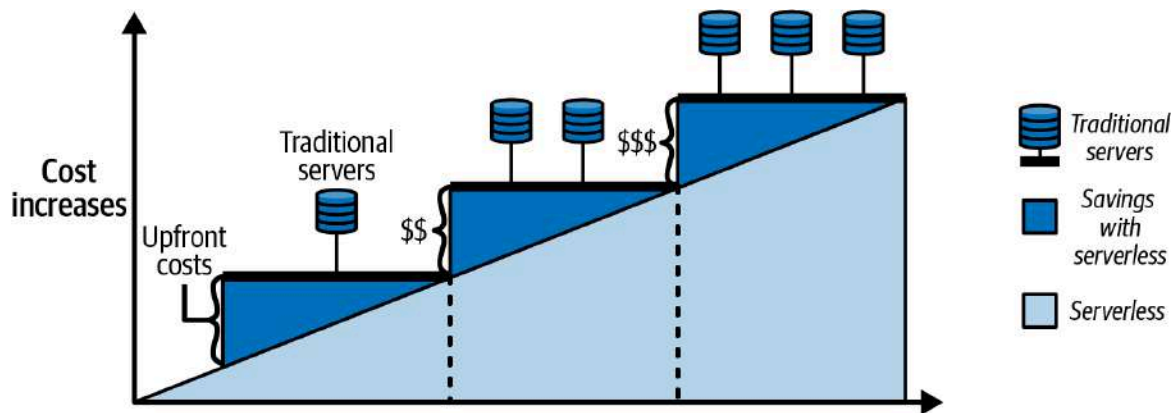


Figure 3-7. How serverless computing helps in cost savings in IT infrastructure

Because serverless providers like Azure provide autoscaling capabilities, they can help reduce the cost of maintaining physical servers or data centers.

Azure Functions

Azure Functions is a powerful compute service for **serverless computing**. Serverless functions in Azure are ideal when you're concerned only about the code running your service and not the underlying platform or infrastructure. For example, you can use Azure Functions for the backend logic of your APIs or integration with other services or APIs. This compute service is event-driven, which means it can be used if you need to do tasks in response to an event, for example an HTTP-triggered message arriving in an Azure Service Bus Queue that passes the queue message to another enterprise application or service. Benefits of Azure Functions include:

- Write less code by focusing on writing the logic
- Focus on development and delivery
- Less work on maintaining the servers and infrastructure
- Save on costs by paying only when functions are running
- Flexibility of using different languages and frameworks
- Autoscaling

- Wide variety of integration options with other services

Components of Azure Functions

Azure Functions is designed for event-driven applications. It has two components:

Triggers

An Azure function will not run without triggers. A trigger based on any event defines how an Azure Function is invoked. It is important to note that an Azure Function must have one trigger and the triggers in a function can have corresponding data that usually serves as the payload.

Bindings

To declaratively connect another resource to the function; bindings are optional and may be connected as input, output, or both.

Both triggers and bindings let you avoid hard-coding access to other services. Your function receives data (content of a queue message) as parameters in your function.

When you create and configure your Azure Function, all triggers and bindings have a specific *direction-property* in the functions configuration file *function.json*. Triggers always have the *in* direction and the bindings use *in* and *out* depending on if it was an input or an output. Another direction called *inout* is supported in some bindings. Learn more in “[Azure Functions Triggers and Bindings](#)”.

CREATE AND BUILD AZURE FUNCTIONS

Azure Functions can be developed in different programming languages like C#, Java, PowerShell, Javascript, Python, Typescript, etc. Try building your first Azure Functions via Microsoft’s “[Getting Started with Azure Functions](#)”.

Azure Durable Functions

Durable Functions is an Azure Functions extension used to develop stateful functions within a serverless computing environment. These functions can be executed without the need for a server. Durable Functions has the same standard functionality as Azure Functions, but its essential purpose is for constructing serverless stateful workflows and orchestration. Creating and developing your logical workflows with the help of orchestrator functions and stateful entities can help solve complicated use cases or scenarios.

Developing stateful workflows with Durable Functions is beneficial in use cases such as automating business processes like approval workflow that require a human response or interaction. For example, when you have a scenario that requires you to wait for an external event. Durable Functions helps keep workflows stateful by managing state and checkpoints and restarts the workflow for you; you do not have to manually code the state management of each function associated with this extension. As a result, developers gain productivity and can focus on writing the code, implementing the business logic, and delivering the business requirements.

As of this writing, Durable Functions supports languages such as C#, JavaScript, Python, F#, and PowerShell. It also supports two types of process models for .NET class library functions: **In-Process and Isolated models**. The **.NET isolated worker** option enables developers to customize their programming by allowing them to use isolation and run their functions app in a different .NET version instead of the default function hosts. It is important to keep up with the latest roadmap updates for Azure Functions. As the .NET version gets upgraded, it will continue to use isolated worker process.

The benefit of using isolated process in Azure Functions for .NET is the advantage of having less conflicts in your functions since each run in a separate process. You also have full control of its process, implement dependency injection for .NET, and can add **middleware** into your function app.

NOTE

The Durable Task Framework (DTFx) is the library behind Azure Durable Functions. DTFx allows users to write long-running (stateful) and persistent workflows (orchestrations) in programming language C#.

Learn more about this framework on [GitHub](#) and watch this interesting video about it: [Building Workflows with the Durable Task Framework](#).

Key features and benefits of Azure Durable Functions

- Take control of complex use case workflows
- Solve event-driven scenarios in your applications
- Develop stateful orchestration workflows of your functions or tasks in serverless environments
- Less maintenance of infrastructure so you can focus on programming your event-driven logic
- Pay only for instances when your function apps are running
- Integration flexibility with other Azure services and APIs

Components of Azure Durable Functions

Azure Durable Functions components or function types play important roles in the serverless stateful workflow and orchestration of tasks. A typical durable function app in Microsoft Azure is composed of different types of Azure Functions. Each of them can be stateful or stateless depending on the type you are using and what it should be doing.

Durable functions have four types: client functions, orchestrator functions, activity functions, and entity functions, as illustrated in [Figure 3-8](#).

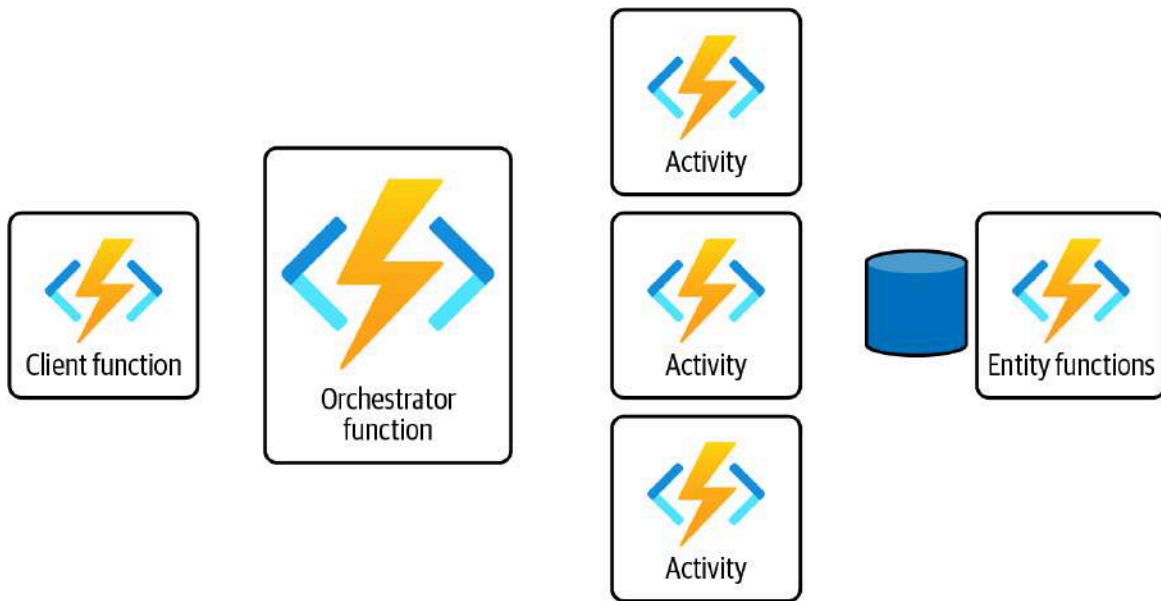


Figure 3-8. Types of Azure durable functions

Let's look at the details of each of these durable function types to better understand how it all works in authoring and developing stateful workflows or orchestration.

Client functions

Client functions start the orchestration the workflow of durable functions. A durable function needs to have an **orchestration client binding** to trigger the instance of the orchestrator functions.

Any Azure function that is non-orchestrator can be a client function. For example, you can trigger and start the orchestration of your workflow using an HTTP trigger, Azure Cosmos DB trigger, Azure Service Bus trigger, and other triggers supported by the Azure Functions framework. The difference is that a client function uses a `DurableClient` output binding as shown in **Example 3-1**.³

Example 3-1. Client function using Azure BLOB trigger to start orchestration of the durable workflow by calling the orchestrator function `StorageOrchestrator`

```
[FunctionName("BlobTriggerStart")]
public static async Task BlobTriggerClientFunction(
    [BlobTrigger("photoscontainer/{name}"),
```



```

        Connection = "StorageConnectionString")]
        CloudBlockBlob myBlob, string name,
        ILogger log, [DurableClient] IDurableOrchestrationClient starter)
    {
        try
        {
            log.LogInformation($"Started orchestration triggered by BLOB trigger.
                                A blob item with name = '{name}'");

            // Function input comes from the request content.
            if (myBlob != null)
            {
                var blobItem = new CloudBlobItem
                {
                    Name = myBlob.Name,
                    BlobUrl = myBlob.Uri.AbsoluteUri.ToString(),
                    Metadata = (Dictionary<string, string>)myBlob.Metadata,
                    FileType = myBlob.BlobType.ToString(),
                    Size = myBlob.Name.Length.ToString(),
                    ETag = myBlob.Properties.ETag.ToString()
                };

                var instanceId = await
starter.StartNewAsync("StorageOrchestrator",
                        blobItem);
                log.LogInformation($"Started orchestration ID =
'{{instanceId}}'.");
            }
            else
            {
                log.LogError($"The blob was triggered but myCloudBlob was empty");
            }
        }
        catch (Exception ex)
        {
            //TODO Error handling
            log.LogError("Something went wrong. Error : " + ex.InnerException);
            throw;
        }
    }
}

```

Orchestrator functions

Orchestrator functions provide an overview of the actions carried out as well as the sequence in which actions are carried out. As seen in the Durable Functions application patterns, orchestration is described by the

orchestrator functions, which are written in code (C# or JavaScript). An orchestration may contain a wide variety of operations, such as activity functions, suborchestrations, HTTP requests, timers, and waiting for external events. The functions of the orchestrator may communicate with the functions of the entity.

NOTE

The code used to write orchestrator functions resembles other types of code; however, there are specific guidelines for how the code should be written. In particular, the code for orchestrator functions needs to be deterministic. If you do not adhere to these determinism requirements, it is possible that the orchestrator functions will not run successfully. There are code constraints developers need to consider that involve in-depth information on these prerequisites as well as instructions on how to get past them.

Learn more here: [“Using Deterministic APIs”](#).

When the orchestrator function gets triggered by any type of client function using the durable orchestration trigger, it starts the orchestration and runs the workflow that you authored and designed in your code.

Example 3-2 shows a simple example of an orchestrator function with a durable stateful workflow by calling the activity functions that do the work. The orchestrator function checks if the BLOB data passed from the client function `BlobTriggerStart` is not empty or of null value before it calls the other activity functions `SendMessageToServiceBusQueue`, `SendSmsCallviaTwilio`, and `SendEmailNotification`.

Example 3-2. An example of an orchestrator function called `StorageOrchestrator` that uses the function chaining pattern

```
[FunctionName("StorageOrchestrator")]
public static async Task<string> RunOrchestrator([OrchestrationTrigger]
    IDurableOrchestrationContext context, ILogger log)
{
    try
    {
        var uploadedBlob = context.GetInput<CloudBlobItem>();
        bool isEmailSentToAdmin;
```

```

//Chain #1 Send Message with BLOB details to Service Bus Queue
var queueMessage = await context.CallActivityAsync<string>(
    "SendMessageToServiceBusQueue",
    uploadedBlob);

if(queueMessage != null)
{
    //Chain #2 Send SMS and call via TwilioAPI
    var isSmsCalledUser = await context.CallActivityAsync<bool>(
        "SendSmsCallviaTwilio",
        serviceBusMessage);

    //Chain #3 send email using Sendgrid API
    if (isSmsCalledUser)
    {
        isEmailSentToAdmin = await context.CallActivityAsync<bool>(
            "SendEmailNotice",
            uploadedBlob);
    }

    log.LogInformation($"A new blob named {uploadedBlob.Name}
    was queued" +
    $"and added to service bus queue. \n" +
    $" SMS sent = {isSmsSentAndCalledUser} to user. \n" +
    $" Access via URL: {uploadedBlob.BlobUrl}" +
    $" and email sent.");
}

return $"Orchestration done with Id: {context.InstanceId}";
}
catch (Exception ex)
{
    //TODO Handle possible errors and do a retry if needed or
    retry a function
    log.LogError($"Something went wrong " + ex.Message);
    throw;
}
}

```

Example 3-2 uses function chaining; however, you can actually combine the different patterns to create a more complex orchestration. Application patterns will be discussed after the components and types of Azure Durable Functions are introduced.

Additionally, in an orchestration, you also have the option to write suborchestrations within your orchestrations. See the [documentation on durable orchestrations](#) to learn more.

Activity functions

Activity functions are the work to be executed in a durable orchestration. The tasks are orchestrated in the series of an event-driven workflow. For example, you might create an orchestrator function to process an order. The tasks involve checking the inventory, sending an email to the customer, creating a shipment process, and other event-driven tasks. These functions can be executed in parallel, or in combination, depending on the type of workflow orchestration pattern you are using for your use case.

These types of durable functions are short-lived functions that perform specific tasks in a workflow. They are called from the orchestrator function, and their input and output are managed by it. Activity functions allow you to break down complex workflows into smaller, reusable tasks and to handle errors and retries more efficiently.

Another simple example from the [Microsoft documentation](#) is shown in [Example 3-3](#). This is an example of the activity functions that receive the parameter of the string of cityName. The HelloGreeter_Activity does its job to output and send a “hello” greeting to the city once it is triggered using the ActivityTrigger. The activity function

SendMessageToServiceBusQueue has the logic to send BLOB data uploadedcloudBlob and compose a string text message to be sent and saved in the Azure Service Bus queue. Once that queue message is saved in the queue storage in Azure, it will return that queue message to the orchestrator function StorageOrchestrator.

Example 3-3. An example of the first task or activity being called by the orchestrator StorageOrchestrator

```
[FunctionName("SendMessageToServiceBusQueue")]
public static async Task<string>
SendMessageToAzureServiceBusQueueAsync([ActivityTrigger]
    CloudBlobItem uploadedBlob, ILogger log,
    ExecutionContext executionContext)
```

```

{
    log.LogInformation($"Received data {uploadedBlob.Name},
        format {uploadedBlob.FileType}.");

    //Config settings for Azure Service Bus
    var azureServiceBusConfig = new ConfigurationBuilder()
        .SetBasePath(executionContext.FunctionAppDirectory)
        .AddJsonFile("local.settings.json", optional: true, reloadOnChange:
true)
        .AddEnvironmentVariables()
        .Build();

    var serviceBusConnection =
    azureServiceBusConfig["AzureServiceBusConnectionString"];
    var serviceBusQueue = azureServiceBusConfig["ServiceBusQueueName"];
    string composedMessage = "";

    try
    {
        if (uploadedcloudBlob != null)
        {
            log.LogInformation($"Composing message to be sent to the queue");

            composedMessage = $"A blob image {uploadedBlob.Name} was added
queue
                                + $"Blob Type: {uploadedBlob.FileType}
</br>"
                                + $"Blob URL: {uploadedBlob.BlobUrl}
</br>"
                                + $"Message sent";

            await using (ServiceBusClient client =
            new ServiceBusClient(serviceBusConnection))
            {
                //Create sender
                ServiceBusSender sender =
client.CreateSender(serviceBusQueue);

                //Create message
                ServiceBusMessage msg = new
ServiceBusMessage(composedMessage);

                //Send Message to ServiceBus Queue
                await sender.SendMessageAsync(msg);
                log.LogInformation($"Sent queue message: {serviceBusQueue}");
                return composedMessage;
            }
        }
    }
}

```

```

        else
        {
            return composedMessage;
        }
    catch (Exception ex)
    {
        log.Error($"Something went wrong");
        log.Error($"Exception {ex.InnerException}");
        throw;
    }
}

```

As soon as the user uploads a blob image to the storage account, a queue message is composed for an Azure Service Bus queue, which will then conduct the tasks of sending an email, call, or SMS to the user or the administrator. This is just a simple implementation; you can do more complex things with Durable Functions.

To learn more about different use cases, check out the list of interesting projects on the [Azure Serverless Community Library](#).

Entity functions

This feature of Azure Durable Functions manages stateful entities in a distributed, scalable, and reliable manner. An entity is an instance of a stateful object that a Durable Function can create, update, and delete.

As a developer, it allows you to define a type of stateful object as a class and annotate it with an attribute. This class can be used to determine the state and behavior of an entity, including its properties, methods, and events. The state of an entity is stored in the Azure Durable Functions state storage, designed to be durable and reliable.

Entity functions manage the state of entities using **optimistic concurrency control**. When a durable function accesses an entity, the current version of the entity state is retrieved from the state storage. The durable function can then modify the entity's state and save the new state back to storage. Suppose another durable function has changed the entity state in the meantime. In that case, the save operation will fail with a concurrency

exception, and the durable function can handle the exception and retry the operation if necessary.

These functions help manage stateful entities in a distributed system, such as users, accounts, orders, or inventory items. Entity functions allow you to encapsulate the logic and state of an entity in a single class, making it easier to maintain and evolve. They also provide a way to ensure consistency and reliability when multiple durable functions access the same entity simultaneously.

The following list summarizes the uses and benefits of using entity functions in Azure Durable Functions:

Scalability

Handles many concurrent requests and scales dynamically to meet demand.

Reliability

Provides durable state storage designed to be highly available and fault-tolerant.

Consistency

Uses optimistic concurrency control to ensure that the state of an entity is consistent and that updates are applied in the correct order.

Encapsulation

Allows you to encapsulate the logic and state of an entity in a single class, making it easier to maintain and evolve.

Reusability

Can be used across multiple workflows and orchestrations, providing a way to reuse familiar stateful entities across different parts of an application.

With entity operations in your function, you can use the following types of operations:

Entity Id

Your target entity

Operation name

Name of the operation to be performed on the entity

Operation input (optional)

Input parameter for the operation

Scheduled time (optional)

Delivery time of the operation

TIP

If you are using .NET for durable entity functions, there are two ways to define your durable entities: *class-based* and *function-based*. The *class-based syntax* usually describes entities and its operations as classes or methods, which is more flexible and readable, especially if you want to use interfaces. On the other hand, *function-based syntax* allows precision control over your entity functions such as how entity state is managed or how entity operations are being dispatched.

Overall, entity functions are a powerful feature of Azure Durable Functions that provide a way to manage stateful entities in a distributed, scalable, and reliable manner. They allow you to encapsulate the state and behavior of an entity in a single class and provide a way to ensure consistency and reliability when multiple durable functions access the same entity simultaneously.

NOTE

Entity functions and related functionality in Durable Functions are only available in **version 2.0 and higher**. They are currently supported in .NET, JavaScript, and Python. As of this writing, the feature is not available in Java or Powershell.

If you want to see the entire source code of these durable function examples, please free to clone my GitHub repository: “**Azure Durable Functions (Function Chaining Example) in C# .NET (Starter Template)**”. You may use it as a template to get started with your own durable functions or as a supplement to this section of the book.

Orchestration Triggers Kickstart Durable Functions

So how do durable functions get triggered?

Durable functions have trigger bindings for execution of the orchestration, activity, and entity functions. Like standard Azure Functions, Durable Functions should never use input or output bindings since adding them will create conflicts and issues with the Durable Task extension. The orchestrator trigger is designed to execute if a new instance of orchestration is being scheduled or when the current instance of the orchestrations receives a new event. Such an event can come from external clients, HTTP triggers, **durable timer events**, etc.

The orchestrator trigger configuration is unique and can be customized depending on the type of programming languages you are using to code your function app. For example, authoring functions in .NET uses the class **OrchestrationTriggerAttribute**. If you are authoring it in PowerShell, JavaScript, or Python, you need to define the configuration for this trigger in the *functions.json* file, as shown in **Example 3-4**. The orchestration property is optional.

Example 3-4. Format of the functions.json file that describes the configuration for a durable function app.

```
{
  "name": "<Name of input parameter in function signature>",
  "orchestration": "<Optional - name of the orchestration>",
  "type": "orchestrationTrigger",
  "direction": "in"
}
```

The orchestrator is deterministic

Of all the Durable Functions types, the orchestrator functions play an important role in keeping your workflow running smoothly. Event sourcing is used to maintain the state and reliability of task execution. The orchestrator should be designed in a deterministic way because the workflow you write in the code will be reused many times. It expects the same result each time. For this reason there are some documented and recommended code constraints to follow when designing the workflow in the orchestrator.

Table 3-3 lists some of the code constraints that you need to know when designing and writing your workflow in the orchestrator function.

Table 3-3. Constraints of writing a workflow orchestration in the orchestrator function

Do not do these	What you can do instead
Avoid generating GUIDs or random numbers	Use the <code>NewGuid()</code> method within the context
Don't access configurations and data	Pass data stores or configuration into activity functions
Never create infinite loops in the orchestrator	<code>CreateAsNew()</code> method in <code>DurableOrchestrationContext</code>
Avoid having blocking APIs and threads (<i>Sleep, Run, Delay</i>)	Use <code>DurableOrchestrationContext</code> to control the state of your tasks
Never use standard <code>CurrentDateTime()</code> method in .NET	Use <code>CurrentUtcDateTime()</code> method in context

There are other important things to consider when you design your stateful orchestration with Azure Durable Functions that fall outside the scope of this book but may be useful for your particular use case. Learn more about the other common orchestrator function code constraints in the [Microsoft documentation](#).

Durable Function Types: Stateful or Stateless?

One of the features that makes Azure Durable Functions effective and useful is its capability to keep the workflow or orchestration of our activity functions stateful. Red Hat's article explains the [difference between stateful versus stateless](#) and the Microsoft article "[Affairs of State: Serverless and Stateless Code Execution with Azure Functions](#)" is also worth checking out

for design considerations and best practices with development for Azure Functions in serverless architectures.

Figure 3-9 illustrates what state each durable function type is. Client and activity functions are like the standard Azure Functions that are stateless. On the other hand, the orchestrator and entity functions are stateful.

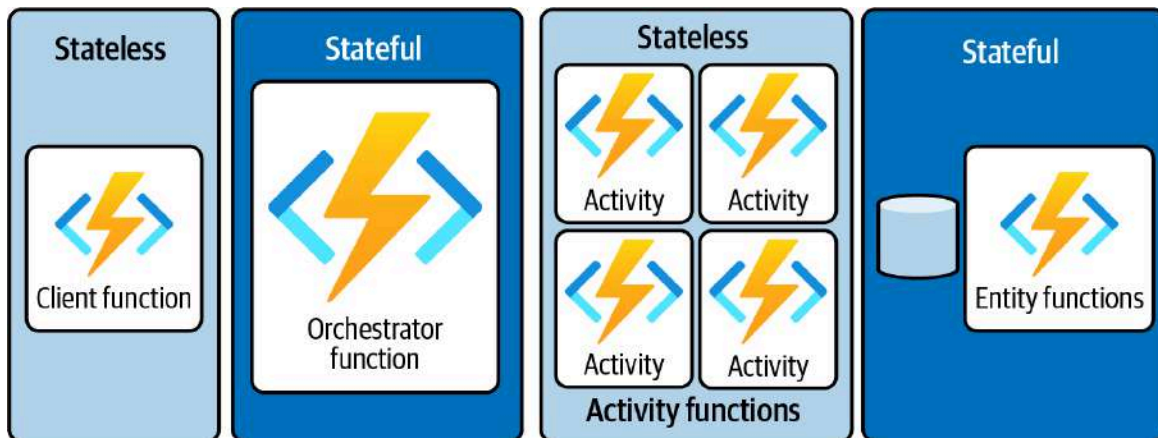


Figure 3-9. The different state for each of the Durable Function types

Application patterns for serverless stateful workflows

One of the key features that makes Azure Durable Functions a great serverless compute service is that it enables us to write stateful workflows in serverless architecture or environment. Stateful workflows and orchestrations help us solve some of the complex problems in our applications. Normally, complex orchestration use cases are difficult to handle by just using normal functions that lose state (stateless).⁴

Six known application patterns have been documented by Microsoft for Azure Durable Functions. These patterns are useful in solving common complex use cases in the serverless architecture that we will explore next.

Function chaining

Figure 3-10 shows an example function chaining pattern. In the function chaining pattern shown in **Example 3-5**, a sequence of functions executes in a specific order like a chain. The output of one function is applied as the input of the next function.

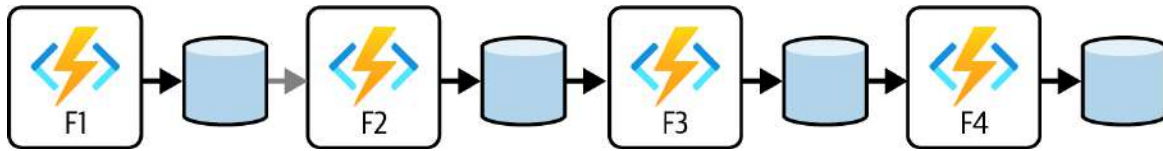


Figure 3-10. An example of the function chaining pattern

Example 3-5. Function chaining pattern. Source: [Azure Durable Functions documentation](#)

```
[FunctionName("FunctionChainingExample")]
public static async Task<object> Run([OrchestrationTrigger]
IDurableOrchestrationContext context)
{
    try
    {
        var res1 = await context.CallActivityAsync<object>("myfunc1", null);
        var res2 = await context.CallActivityAsync<object>("myfunc2", res1);
        var finalresult = await context.CallActivityAsync<object>("myfunc3",
res2);
        return await context.CallActivityAsync<object>("myfunc4",finalresult);
    }
    catch (Exception)
    {
        // Error handling, activities or retry-functions to handle exception
    }
}
```

Function chaining is a Durable Functions pattern allowing you to execute a series of functions in a specific order. In this pattern, each function in the chain depends on the output of the previous function, forming a chain of process calls that pass data from one function to the next.

This pattern is useful when you need to break down a complex task into smaller, independent steps and ensure that the steps are executed in the correct order. This pattern can be used in various scenarios, such as processing a batch of files, performing a series of calculations, sending a sequence of messages, and other tasks that require sequence processing.

This pattern is achieved through an orchestration function that calls a series of activity functions. The orchestration function is responsible for managing the state of the workflow and calling the activity functions in the correct order.

The basic workflow of the function chaining pattern in Azure Durable Functions is as follows; it is also illustrated in [Example 3-5](#).

1. The workflow starts with an orchestration client function that initializes the state of the orchestration workflow and calls the first activity function in the chaining pattern.
2. The activity function performs its task and returns the output to the orchestrator function.
3. The orchestrator function uses the output of the previous activity function to determine which activity function to call next.
4. The process repeats until all activity functions in the chain have been executed.
5. The final output of the last activity function is returned as output of the orchestration workflow.

When an activity function is called, it executes as a separate Azure Function and returns its result to the orchestrator function. If the activity function fails or times out, the orchestrator function can handle the error and retry the activity function if necessary.

The function chaining pattern in Durable Functions is a method to break down difficult tasks into smaller, independent steps and ensure that the steps are executed in the correct order. It provides a way to manage the state of the workflow and handle errors and retries efficiently.

Fan-out / fan-in

Fan-out/fan-in is a pattern in Azure Durable Functions that lets you execute multiple functions at the same time and aggregate the result. In this pattern, various functions process a single input in parallel, and the resulting output is combined. You code the workflow to execute multiple functions in parallel. Then you wait for all functions to finish, as shown in [Example 3-6](#). Usually the aggregation work is done on the results that are returned from the functions, as illustrated in [Figure 3-11](#).

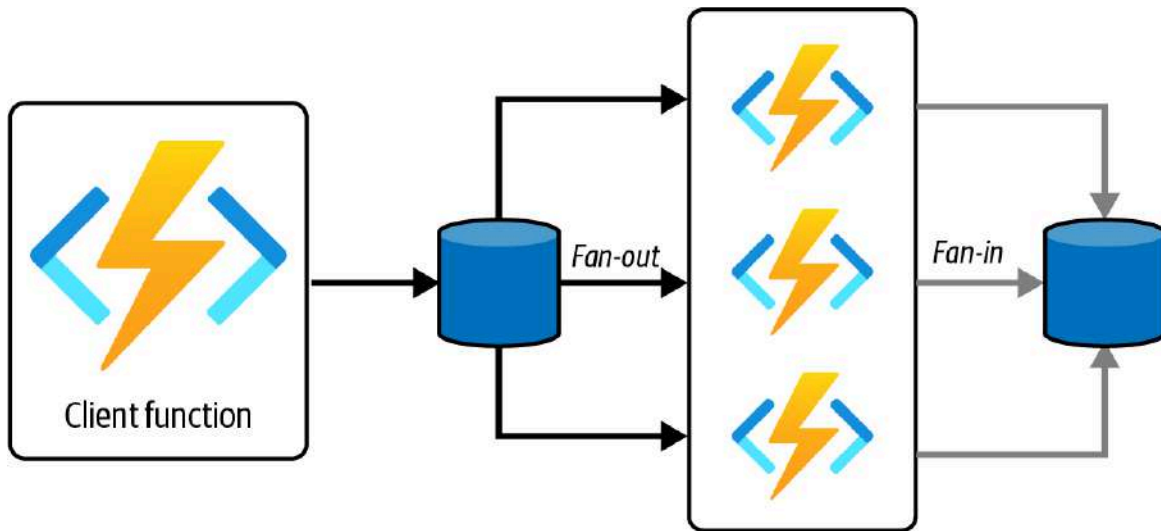


Figure 3-11. An example of the fan-out/fan-in pattern

This orchestration pattern is advantageous when a large quantity of data must be processed or multiple independent tasks must be completed concurrently. This pattern is applicable in various situations, including data processing, batch processing, and parallelizing compute-intensive tasks.

Example 3-6. Fan-out/fan-in pattern. Source: [Azure Durable Functions documentation](#)

```

[FunctionName("FanOutFanInExample")]
public static async Task Run([OrchestrationTrigger]
IDurableOrchestrationContext context)
{
    var parallelTasks = new List<Task<int>>();

    // Get a list of N work items to process in parallel.
    object[] workBatch = await context.CallActivityAsync<object[]>("F1", null);
    for (int i = 0; i < workBatch.Length; i++)
    {
        Task<int> task = context.CallActivityAsync<int>("F2", workBatch[i]);
        parallelTasks.Add(task);
    }

    await Task.WhenAll(parallelTasks);

    // Aggregate all N outputs and send the result to F3.
    int sum = parallelTasks.Sum(t => t.Result);
    await context.CallActivityAsync("F3", sum);
}
  
```

In this pattern, the orchestration function is responsible for administering the workflow's state and coordinating the execution of activity functions in parallel.

A workflow starts with an orchestration function that initializes the workflow's state and prepares input data for parallel processing. Using the `CallActivityAsync` method, the orchestration function calls multiple activity functions in parallel and transmits a partition of the input data to each activity function. Each activity function executes the assigned task autonomously and returns the result to the orchestrator function. The orchestrator function awaits the completion of all activity functions using the `Task.WhenAll` method. The orchestrator function returns a single output containing the outcomes of all activity functions.

To implement this pattern in Durable Functions, you must define the activity functions as distinct and decorate them with the `[FunctionName]` attribute. You then contact them in a loop from the orchestration function using the `CallActivityAsync` method, passing each activity function a partition of the input data.

When an activity function is invoked or triggered, it runs as its own Azure Function and returns its result to the orchestrator function. The orchestrator function can manage the error and retry the activity function if the activity function fails or times out.

Overall, this pattern is an effective method that executes multiple functions in parallel and aggregates their results into a single output. It provides a way to manage a workflow's state while effectively handling errors and retries.

Async HTTP APIs

If you want to solve problems related to state coordination of long-running operations with external clients or APIs, then the Async HTTP APIs pattern might be useful.

A common way to implement the Async HTTP API pattern is by having the long-running action be triggered by an HTTP call, and then redirecting the

client to a status endpoint that it can poll to learn when the operation completes. **Figure 3-12** shows the pattern starting, doing the work, and getting the status over HTTP.

NOTE

Durable Functions provide built-in APIs that simplify the code we write for interacting with long-running function executions.

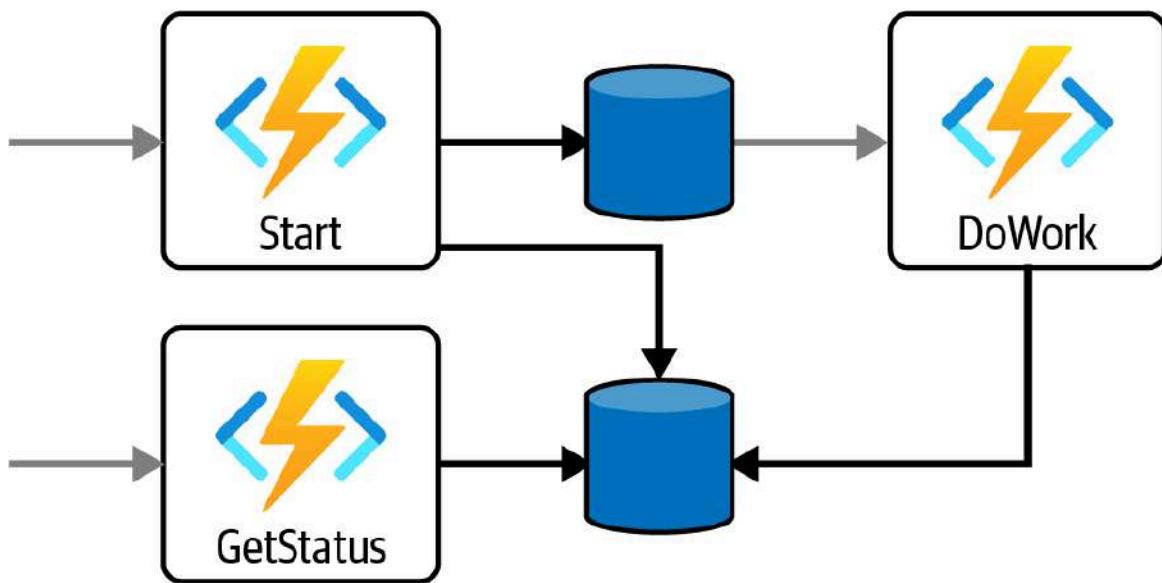


Figure 3-12. Async HTTP APIs pattern

Example 3-7 shows a simple way we can use the Async HTTP APIs pattern to make a HTTP request to any URL or endpoint.

Example 3-7. Async HTTP API pattern checking website status and content via HTTP

```
[FunctionName("CheckWebsiteContentFunction")]
public static async Task CheckWebsite([OrchestrationTrigger]
    IDurableOrchestrationContext context, ILogger log)
{
    try
    {
        Uri jonahsWebsite = new Uri("https://jonahandersson.tech");

        //Make an HTTP request ex. GET or POST to specific endpoint or URL
```

```

DurableHttpResponse httpResponse =
await context.CallHttpAsync(HttpMethod.Get, jonahsWebsite);

if((int)httpResponse.StatusCode >= 500
&& (int)httpResponse.StatusCode < 600)
    throw new HttpRequestException("Hey, there is a server error!");
else
    log.LogInformation($"Completed");
    log.LogInformation($"Response Code: {httpResponse.StatusCode}");
    log.LogInformation($"Content: {httpResponse.Content}");
}
catch (Exception)
{
    // Error handling, activities or functions to handle exception.
}
}

```

Finally, if you want to learn more about how you can use the Durable Functions Async HTTP APIs pattern to expose asynchronous, long-running processes over HTTP, check out this [Microsoft documentation](#) about its [HTTP features](#).

Monitor pattern

In context with a workflow, the monitoring pattern relates to an adaptable iterative procedure, such as polling until specific requirements are satisfied. A simple case, such as a recurring cleanup task, can be handled by a regular timer trigger; however, because its interval is fixed, it makes managing the instance lifetimes more difficult.

Durable Functions make it possible to construct numerous monitor processes from a single orchestration and have adjustable recurrence intervals and job lifespan management.

An example use case where the monitor pattern can be useful is when you have to run monitoring job processes, as illustrated in [Example 3-8](#).

Example 3-8. Recurring monitoring job workflow using monitor pattern

```

[FunctionName("MonitorCleanUpJobStatus")]
public static async Task Run([OrchestrationTrigger]
IDurableOrchestrationContext context)
{
    int cleanUpJobId = context.GetInput<int>();

```

```

int jobPollingInterval = GetPollingInterval();
DateTime expirationTime = GetExpiryTime();
string emailAddress = GetEmailAddress();

while (context.CurrentUtcDateTime < expirationTime)
{
    // Monitor job status by job id.
    var status =
    await context.CallActivityAsync<string>("GetJobCleanUp", cleanUpJobId);
    if (status == "Completed")
    {
        // Perform an action when a condition is met.
        await context.CallActivityAsync("SendMonitoringAlert", emailAddress);
        break;
    }

    // Orchestration sleeps until this time.
    var nextMonitorCheck =
    context.CurrentUtcDateTime.AddSeconds(jobPollingInterval);
    await context.CreateTimer(nextMonitorCheck, CancellationToken.None);
}

// Perform more work here, or let the orchestration end.
}

```

So, the monitor pattern can be useful in implementing tasks or monitoring jobs that are recurring and need to be stateful. You can also combine it with a durable timer to control the time and workflow as you prefer.

Human interaction

Some processes and tasks require some kind of human interaction. The tricky thing about involving us humans in an automated process is that we are not always available 24/7 and responsive like computer systems or cloud systems. For this reason, developing solutions for the automation of processes can be beneficial. Often timers and programming logic are used to do this, as shown in the example of approval workflow in [Figure 3-13](#) and [Example 3-9](#).

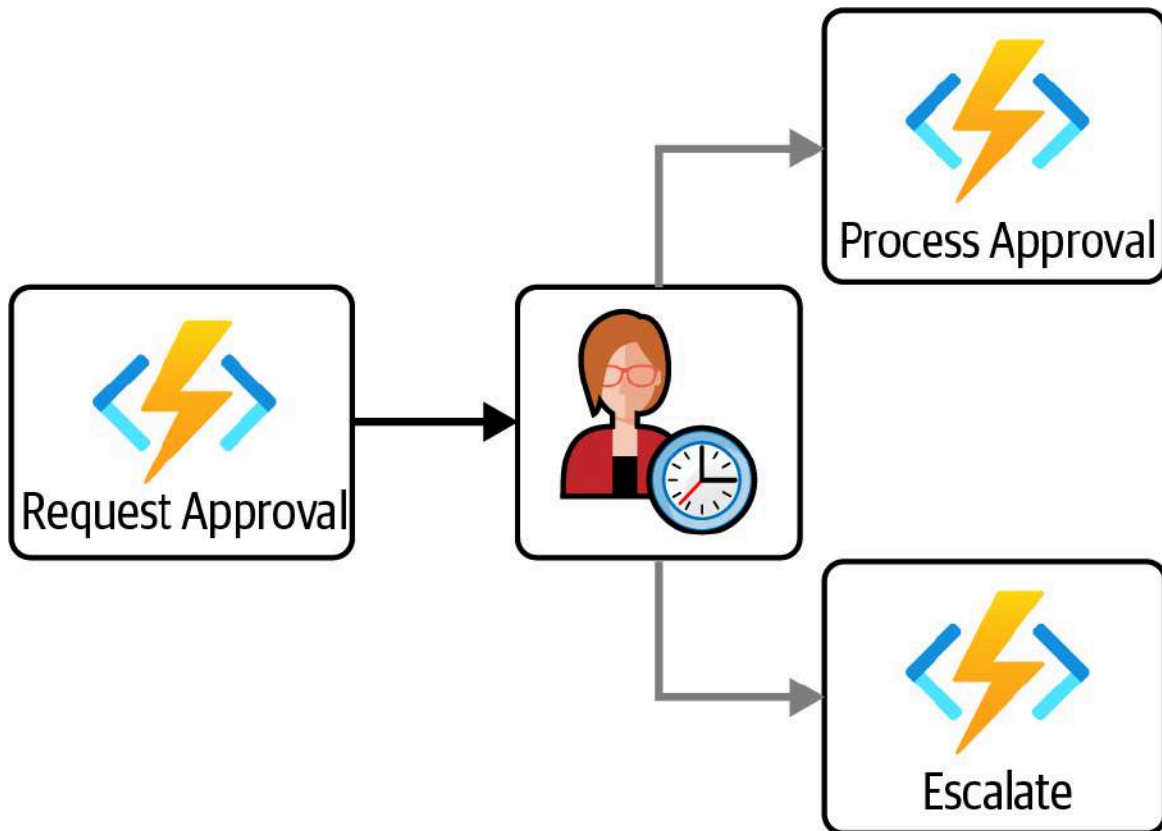


Figure 3-13. Human interaction pattern

Example 3-9. Simple manager approval workflow coded in C# using human interaction pattern

```

[FunctionName("ApprovalFromManagerWorkflow")]
public static async Task Run([OrchestrationTrigger]
IDurableOrchestrationContext context)
{
    await context.CallActivityAsync("RequestApprovalFromManager", null);
    using (var timeout = new CancellationTokenSource())
    {
        DateTime dueTime = context.CurrentUtcDateTime.AddHours(72);
        Task durableTimeout = context.CreateTimer(dueTime, timeout.Token);

        Task<bool> humanInteractionEvent =
            context.WaitForExternalEvent<bool>("ManagerApprovalEvent");
        if (humanInteractionEvent ==
            await Task.WhenAny(humanInteractionEvent, durableTimeout))
        {
            timeout.Cancel();
            await context.CallActivityAsync("ProcessRequest",
                humanInteractionEvent.Result);
        }
    }
}
  
```

```

    else
    {
        //If there is no approval or it timed out
        await context.CallActivityAsync("EscalateToAnotherPerson", null);
    }
}
}

```

Aggregator pattern

If you want to collect or aggregate event data over a specific period of time into a single, transmittable entity, then the aggregator pattern is ideal for your durable orchestration. The event data being collected and aggregated can be from several resources, which can also be delivered into batches or groups. These event data may be scattered over time. The aggregator is designed to take action when the event data arrives and those external services or clients might need to query the event data that is being aggregated.

The aggregator pattern is not easy to implement in normal functions that are not stateful because of the challenge of concurrency problems. In normal stateless functions, you need to handle the concurrent multiple threads with the same data and also make sure that the aggregator runs on a single VM at a time.

There are **code examples** to implement the aggregator pattern for stateful entities.

These different application patterns of Durable Functions are useful in different scenarios depending on the complex problems or use cases you are trying to solve in the serverless or backend part of your applications. If you want to learn more about real examples, consult these publications about the use of serverless and Azure Durable Functions:

- “**Durable Functions: Semantics for Stateful Serverless**” by Burckhardt et al.
- “**Serverless Workflows with Durable Functions and Netherite**” by Burckhardt et al.

Container Services in Azure

One of the ways to modernize applications for a cloud platform like Microsoft Azure is to build and develop them using containers. One of the major benefits of using containers is Azure is cost savings for organizations that want to lift-and-shift their existing applications to the cloud without making huge changes. Containerizing existing applications or building them into microservices can help deliver value to the business and application users.

Applications that are containerized have great capabilities for development and CI/CD with fully managed container management using services like Azure Kubernetes. Containers can also be fully controlled and integrated with Microsoft Entra ID for user access management and security.

Azure Containers and Azure Kubernetes Service

Container Instances and Azure Kubernetes Service are Azure compute resources that you can use to deploy and manage containers. Containers are lightweight, virtualized application environments. They're designed to be quickly created, scaled out, and stopped dynamically. You can run multiple instances of a containerized application on a single host machine.

Azure Container Registry

Azure Container Registry (ACR) is a cloud-based managed service that enables users to store, administer, and deploy container images with Azure Kubernetes Service (AKS) and other container orchestrators. ACR serves as a private registry that enables users to store and administer Docker images in a secure and scalable manner.

ACR enables users to rapidly deploy and administer containers in the cloud using the Docker CLI tools they are already familiar with on their local machines. ACR supports Windows and Linux containers, enabling integration with other Azure services, including Azure DevOps, Azure Container Instances, and Azure Kubernetes Service.

You can also use **Azure Container Registry Tasks (ACR Tasks)** to build your container images in Azure on-demand, or automate builds triggered by source code updates, or updates to a container's base image, or timers.

ACR offers a variety of features, such as role-based access management (RBAC), geo-replication, container image scans, and integration with Microsoft Entra ID. RBAC permits users to manage access to their container images by granting permissions to specified users and groups. Geo-replication allows users to replicate container images across numerous Azure regions for increased availability and resiliency. Users can scan their container images for vulnerabilities and security issues using image scanning.

TIP

You can create an Azure Container Registry instance in different ways using the **Azure CLI**, **Azure Portal**, **PowerShell**, **Bicep**, and **ARM templates**.

ACR is a powerful tool that enables users to securely and scalably store, administer, and deploy images of containers in the cloud. For best practices, it's worth deploying ACR in the same Azure region where your containers are deployed to minimize network latency. If you are deploying in multiple regions, activating the geo-replication for ACR is also a good practice to consider.

Azure Container Instance

Deploying applications and systems in containers is one of the common app modernization technologies today. Many organizations are considering **containerization** for many reasons including the benefits of convenient packaging and deployment of applications into containers to the cloud.

Azure Container Instances (ACI) allows us to run containers in Azure instead of running and managing virtual machines for our applications, as

illustrated in **Figure 3-14**. ACI allows you to run and isolate applications and implement task automation.

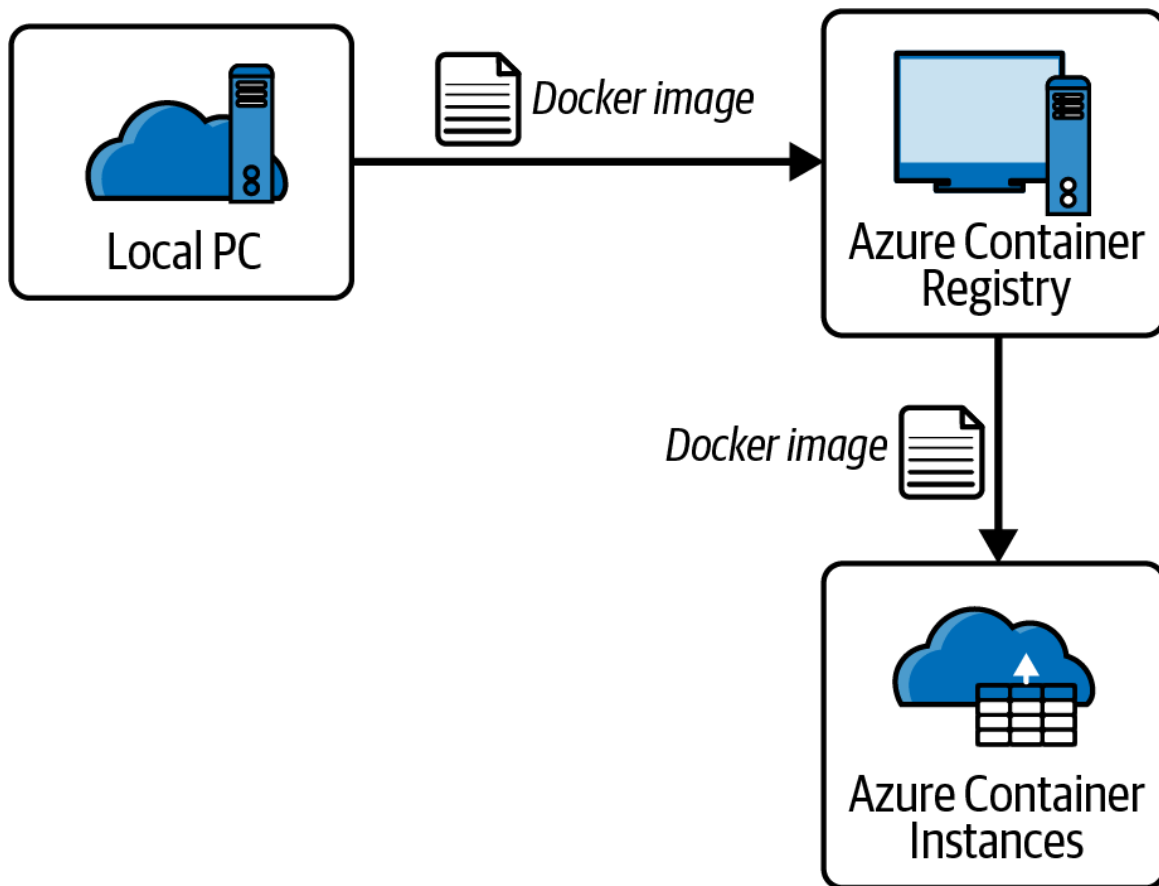


Figure 3-14. Azure Container Instances

Azure Container Apps

Azure Container Apps (ACA) is a compute service designed for developing and deploying applications and microservices using serverless containers. Features include containerized apps without managing complex infrastructure and also support for **built-in authentication** for security.

Although it is not required to use the built-in authentication and authorization in ACA, it is useful because you use the variety of common authentication and **identity providers** including third-party providers, e.g., Microsoft Identity Platform, Microsoft Entra ID, Google, Twitter, GitHub, etc.

ACA combines the benefits of serverless and PaaS and provides a platform for multiple container applications in a serverless environment as shown in **Figure 3-15**. This container solution in Azure has an application lifecycle in three different phases, i.e., deployment, update, and deactivate, based on **revisions**.

NOTE

Aside from the different phases of the application lifecycle mentioned, Azure Container Apps also has a shutdown phase. This phase is triggered when the container app is being deactivated, deleted, or scaled for updates.

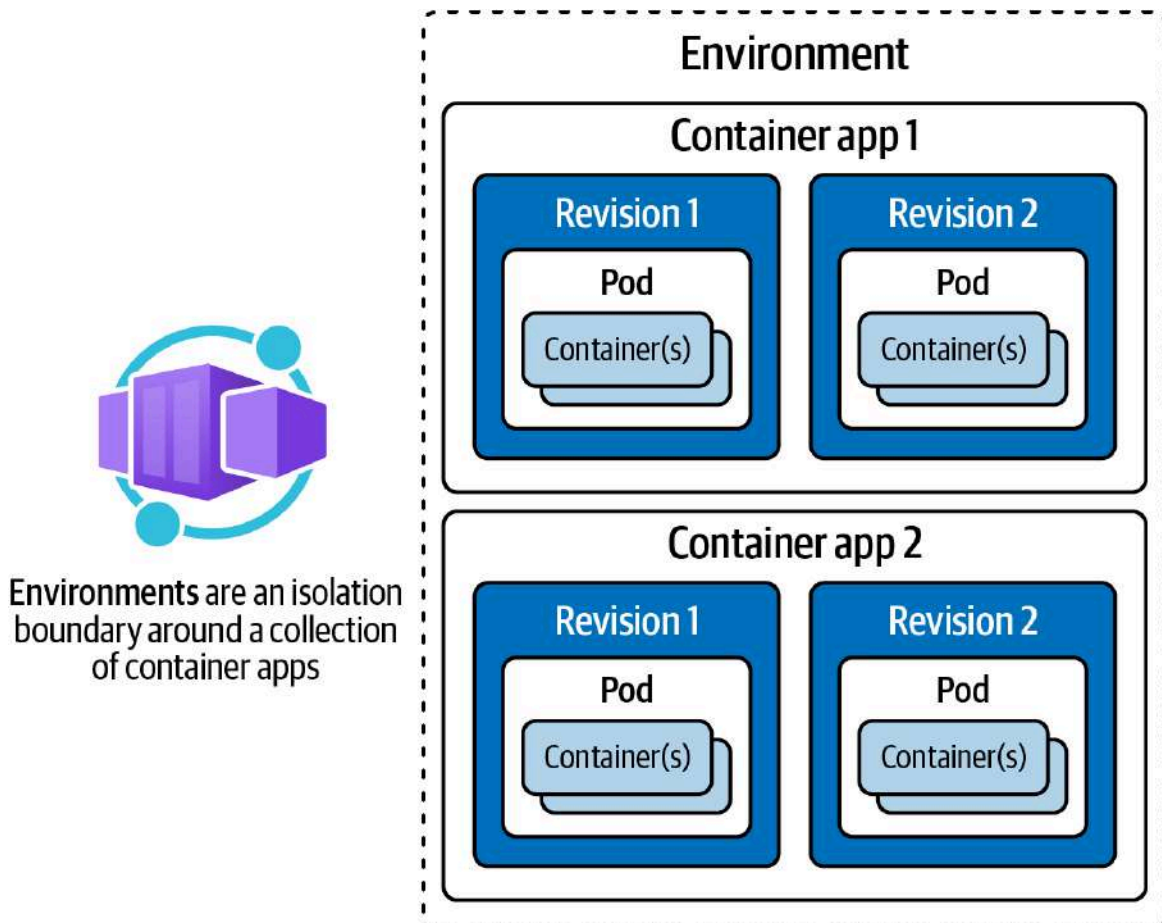


Figure 3-15. The Azure Container Apps environment with container apps or microservices

There are different **container services in Azure** but the benefits and uses shown in **Table 3-4** summarizes how ACA differs from the others.

Table 3-4. Common uses and benefits of Azure Container Apps

Uses and benefits	Description
Running applications in microservices	Develop, deploy, and manage your microservices applications in serverless containers with Distributed Application Runtime (Dapr) integration
Deploying API endpoints	The HTTP traffic can be split into different versions of an application with autoscaling features based on concurrent HTTP traffic or requests
Hosting applications with background processing	Run stateful and long-running background jobs or tasks with autoscaling capabilities based on CPU or memory usage
Managing event-driven processing	Manage and autoscale event-driven processes

Implementing ACA means that you can build containerized applications code using any preferred programming language or framework, build microservices, and have the option to use it with technologies like **Distributed Application Runtime (Dapr)**, **Kubernetes Event-Driven Autoscaling (KEDA)** and **Envoy**.

Figure 3-16 gives an overview of Azure Container Apps and its features along with integration options for Dapr, KEDA, Envoy, and the Azure Kubernetes Service (AKS).

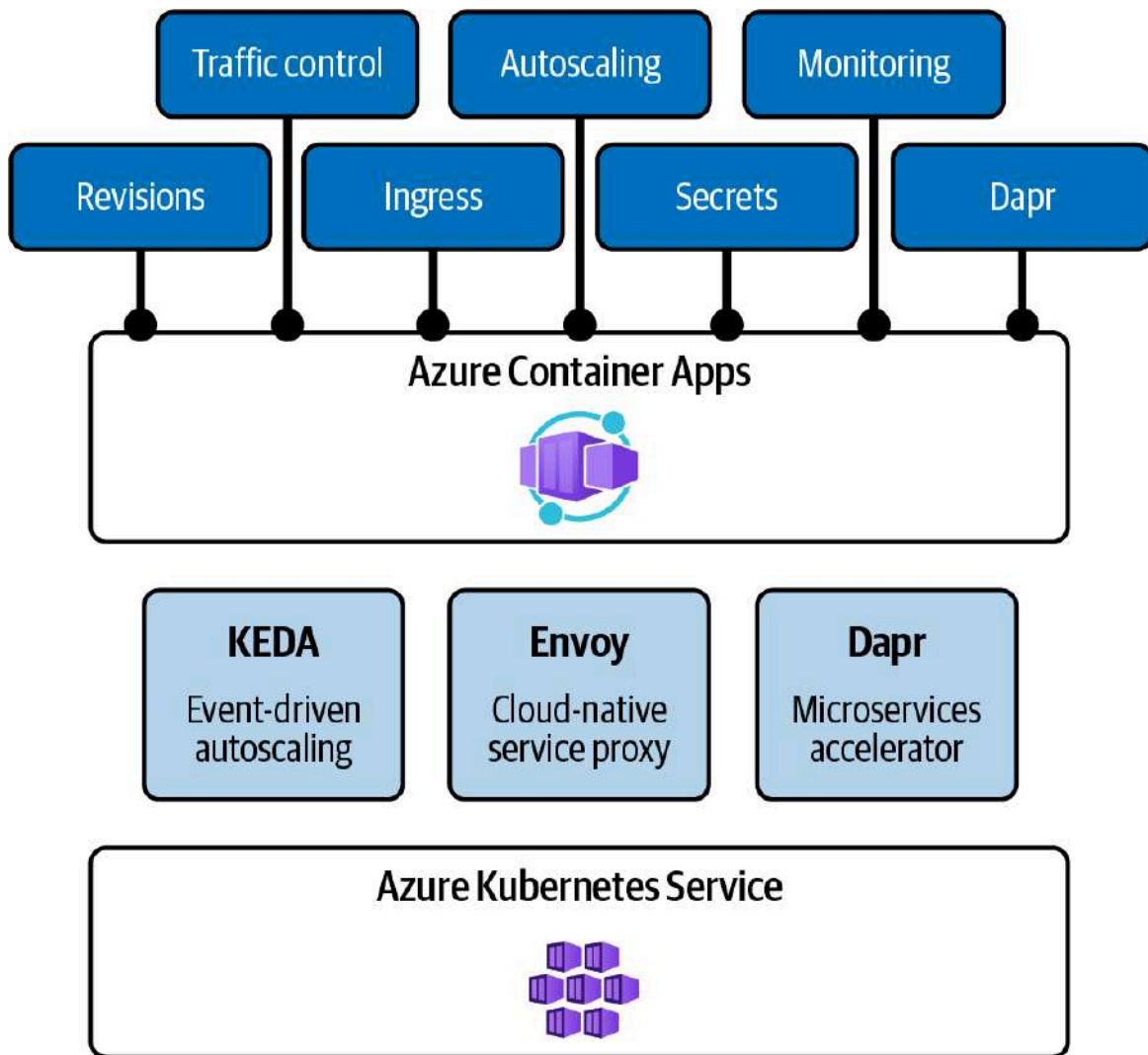


Figure 3-16. An example illustration of Azure Container Apps

Learn more about KEDA by watching [this YouTube video](#). You may also go through the lessons on the [Microsoft Learning Path](#).

If you are also curious about the official KEDA documentation, visit the official [GitHub page of KEDA](#) for code examples and also check out [KEDA scalers](#) that help you detect if a deployment should be disabled or enabled.

NOTE

KEDA is a Kubernetes-based Event Driven Autoscaler. KEDA allows you to manage and control the scaling of any container in Kubernetes according to the number of events that need to be processed. Using **Azure Container Apps with KEDA** allows you to automatically scale your applications down to 0 based on HTTP traffic or requests, CPU, or memory or any of KEDA's event-driven scalers like Azure Service Bus, Azure Monitor, Azure Event Hub, SQL DB, Redis and more.

Aside from using KEDA, you can also deploy an Azure Container App from **Azure Container Registry (ACR)** using Azure CLI and set up CI/CD using Azure DevOps Pipelines.

Furthermore, ACA is a top-level Azure resource, which means that you can develop it with **Infrastructure as code (IaC)** methods using ARM templates, Azure Bicep, Azure CLI, PowerShell, and other external tools like Terraform.

Azure Kubernetes Services

Kubernetes (K8s) is open source orchestration software for deploying, managing, and scaling containers. Azure Kubernetes Service (AKS) is ideal if you have use cases where you need full container orchestration. AKS also includes coordinated application upgrades, automatic scaling, and a feature to perform service discovery across multiple containers.

AKS also offers automated upgrades, self-healing, monitoring, and scalability, making it an ideal solution for deploying and managing containerized cloud applications.

Among the essential uses of AKS in cloud development are:

Container orchestration

AKS simplifies the deployment and management of containerized applications using Kubernetes for container orchestration. It provides a platform that automates

container deployment, scaling, and administration, making large-scale container deployments simple to manage.

Hybrid containers

AKS can deploy containerized applications on premises, in the cloud, or in a hybrid infrastructure. This facilitates the creation of a deployment environment that is consistent across multiple locations.

Container security

AKS offers a secure platform for operating containerized applications, including network security, role-based access control, and container isolation. This ensures that applications are compliant with industry standards.

Support for CI/CD and DevOps lifecycle

AKS can be integrated with Azure DevOps, GitHub, and Jenkins, among other DevOps tools, to automate the deployment, testing, and monitoring of containerized applications. This allows developers to concentrate on application development while AKS manages the underlying infrastructure.

Additional advantages of utilizing AKS include enabling developers to rapidly and easily scale containerized applications based on demand. This ensures that applications are accessible and responsive to user requests at all times.

Figure 3-17 shows a simple example of a use case of AKS in a microservices architecture.

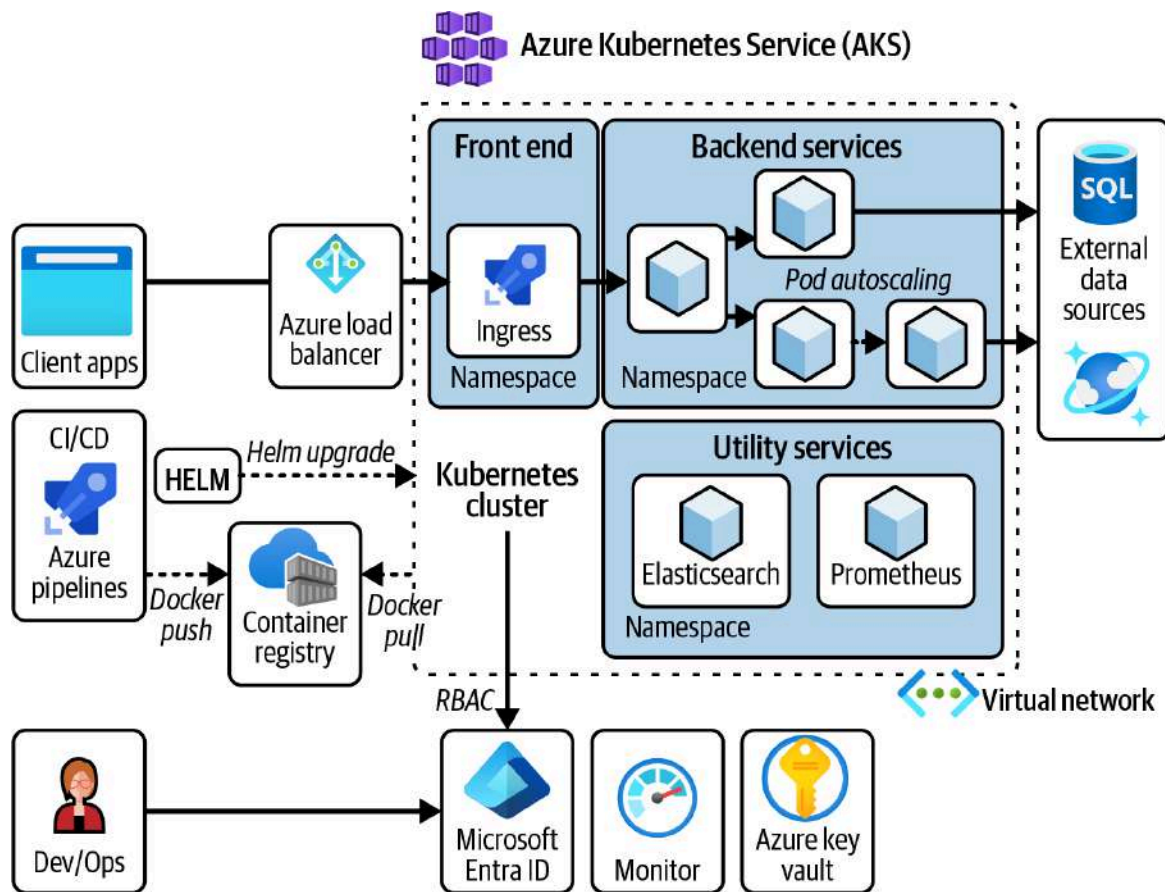


Figure 3-17. Microservices architecture on Azure Kubernetes Service

AKS can be a cost-effective method for running containerized applications, removing the need to administer and maintain the underlying infrastructure. It enables you to implement container technologies on the cloud for operating containerized applications, complete with automatic failover and self-healing capabilities.

TIP

Microsoft Azure has free [learning resources](#) that you can check out to get started with Azure Kubernetes, distributed systems, and cloud-native computing architecture.

Additionally, AKS features capabilities such as [node auto-repair](#), which can detect when a node is unhealthy and repair it for you.

Making Sense of Quantum Computing in Azure

Quantum computers are controllable quantum mechanical devices that exploit the properties of quantum physics to perform computations.⁵ Quantum computing provides exponential advantages when it comes to increasing the speed of some computational tasks. These advantages are possible because of quantum mechanics: superposition, interference, and entanglement. The difference between standard computers and quantum computers is the applications in quantum computers are probabilistic. In comparison, the computers we use daily are usually deterministic. Each possible result produced by quantum algorithms has an associated probability amplitude.⁶

To learn more, read about [quantum computing history in the Microsoft documentation](#).

NOTE

Quantum computers are useful in solving problems that require complex calculations with various possible combinations, for example: cryptography, mechanical systems simulation, machine learning, and algorithms. To try an example yourself, check out this [Quantum Random Number Generator in Q#](#).

Azure Quantum

Azure Quantum is the quantum cloud platform of Microsoft. It is a cloud service with a diverse set of quantum solutions and technologies used to solve quantum computing problems. Even though quantum computing is a new technology, you can prepare for the future, by starting to write your code at the algorithm level.

Azure Quantum Development Kit

Azure Quantum Development Kit (Azure QDK) is used for development with Azure Quantum using Q#, a quantum-focused programming language. Build and run Q# programs on quantum hardware or formulate solutions that execute optimization solvers running on classical hardware on Azure.

You can explore more with Microsoft's quantum learning resources like the [Quantum Computing Foundations Learning Path](#) and join the [Q# Community](#).

Learn by Doing (Try It!)

Aside from the [supplementary hands-on lab repository](#) for the topics in the this chapter, the following quick-start tutorials are recommended as they are updated based on Microsoft's technical updates for the service.

- [Microsoft Quickstart Tutorial for creating a Windows VM in Azure Portal](#)
- [Microsoft Quickstart Tutorial for creating a Linux VM in Azure Portal](#)
- [Microsoft Quickstart Tutorial for hosting a web application using Azure App Service](#)
- [Microsoft Quickstart Tutorial on how to deploy and run a containerized web app with Azure App Service](#)
- [Choose your desired programming language by following through the serverless lessons at Azure Functions University on GitHub](#)
- [Microsoft Quickstart Tutorial for deploying a Dapr Application to Azure Container Apps using Azure CLI](#)
- [Microsoft's Tutorial for deploying a Dapr Application to Azure Container Apps using an ARM or Bicep template](#)
- [Quickstart: Solve an optimization problem in Azure Quantum](#)

Summary

In this chapter, we explored Azure compute services, a widely used category of services in Azure that enables the creation of modern applications and systems. Azure compute services support various architectures such as serverless, microservices, and containers, and provide benefits such as autoscaling, consumption-based pricing, and enhanced developer productivity. We also learned about Azure Durable Functions, which allows the creation of serverless stateful workflows and orchestrations through code while adhering to certain authoring and coding constraints.

Azure App Service and Azure Static Web Apps are tools for empowering modern app development. Azure Functions and Durable Functions are powerful backend services and function apps with event-driven features for stateful workflow orchestrations. Azure Kubernetes, Azure Container Registry, and Azure Container Apps are solutions for creating container apps with orchestration, scaling, and load-balancing capabilities. Additionally, you learned about Azure Quantum, a service and quantum SDK for creating cloud solutions that leverage quantum computing to solve complex problems.

Overall, Azure compute services provide a robust suite of tools for developers, teams, and organizations to build and deploy more scalable, efficient, and secure applications and systems.

Check Your Knowledge

1. What is the difference between Azure VMs and Azure VM Scale Sets?
2. What are Azure Spot VMs?
3. Why are Azure Container Apps the best way to deploy microservices?
4. What Azure compute service would you use to write stateful serverless workflows?

5. What programming language and developments tools would you primarily need for Azure Quantum?

For the answers to these questions, see the [Appendix](#).

Recommended Resources

“App Service Documentation.” Microsoft Learn, <https://oreil.ly/uTJaQ>.

“Azure Container Apps Documentation.” Microsoft Learn, <https://oreil.ly/TDjIi>.

“Azure Container Registry Documentation.” Microsoft Learn, <https://oreil.ly/elDI0>.

“Azure Durable Functions Documentation.” Microsoft Learn, <https://oreil.ly/IqUqP>.

“Azure Functions Documentation.” Microsoft Learn, <https://oreil.ly/EEuWF>.

“Azure Quantum Service Documentation (Preview).” Microsoft Learn, <https://oreil.ly/b5Q3J>.

“Choose an Azure Compute Service.” Microsoft Learn, <https://oreil.ly/3pJgC>.

“Cloud Design Patterns.” Microsoft Learn, April 13, 2023, <https://oreil.ly/caZDI>.

“Developers Guide to Durable Entries in .NET.” Microsoft Learn, October 24, 2023, https://oreil.ly/OQV_u.

“Kubernetes Core Concepts for Azure Kubernetes Service (AKS).” Microsoft Learn, May 2, 2023, <https://oreil.ly/8-jEJ>.

“Learn Kubernetes Basics.” Kubernetes documentation, <https://oreil.ly/UZDOQ>.

Polkovnikov, Alexey. “Azure Services Overview.” Azure Charts, <https://oreil.ly/Rtv8->.

-
- ¹ RedHat.com, “What is virtualization?” <https://www.redhat.com/en/topics/virtualization/what-is-virtualization>
 - ² W3.org, “CORS Enabled?” https://www.w3.org/wiki/CORS_Enabled
 - ³ This example is from one of the serverless development demo projects I published on my [GitHub repository](#) for one of the application patterns, function chaining, that will be discussed later in this chapter.
 - ⁴ Mikhail Shilkov. 2018, “Making Sense of Azure Durable Functions,” <https://mikhail.io/2018/12/making-sense-of-azure-durable-functions/>
 - ⁵ Michael Tabb, Andrea Gawrylewski, and Jeffery DelViscio. 2021. “How Does a Quantum Computer Work?” *Scientific American*, <https://www.scientificamerican.com/video/how-does-a-quantum-computer-work>
 - ⁶ IBM.com, “What is quantum computing?” <https://www.ibm.com/topics/quantum-computing>

Chapter 4. Microsoft Azure Cloud Networking

Azure networking is essential to any public cloud; it allows us to bring both on-premise and cloud networks together. We can scale resources to meet our demands as well as protect our infrastructure; this gives us the flexibility to change to meet any resource demand that our organization or customer may need. With Azure networking, we get compliance and security. It also saves on costs and time which gives us the flexibility to adapt and meet our demands, both current and future.

— Ryan O’Connell, IT Solutions Architect, Microsoft
Azure MVP, Microsoft Certified Trainer, IT Manager,
Blogger at RockITWorks

In **Chapter 3**, you learned about Azure compute services and their benefits for application development in Microsoft Azure. In this chapter we will deep dive into the concepts you need to learn to implement these compute services with the networking services in Azure. By the end of this chapter, you will have learned how you can use the appropriate networking service with existing applications. This will also help in planning networking and hybrid solutions in Azure.

Azure Networking

Azure Networking is a category of services in Microsoft Azure that provides fully managed and scalable networking and connectivity options like making a connection between your on-premises data center and the cloud. With networking services in Azure, you can also build secure **virtual network infrastructure**, manage your application’s network traffic, and protect your applications against **DDoS attacks**. Networking resources in Azure can also be used to enable secure remote access to internal resources

within your organizations and globally route your network connectivity with monitoring and security features.

Using your Azure subscription, while logged in, you can view all of the Azure resources you can create in the networking category on the Azure Portal. You just need to click “Create a resource” and select the Networking category on the **Azure Marketplace** and there you can explore the rest of the networking resources offered by Microsoft and Microsoft Partners.

Azure Networking Services Categories

There are many networking services to choose from in Azure; they are categorized according to their purpose.

Services for connectivity

You can use these Azure networking resources to create and build connectivity-related solutions in the cloud. For example, if you want to connect your Azure resources to your on-premises resources, you can use Azure Virtual Network (Azure VNet), ExpressRoute, Virtual WAN, Virtual Network NAT Gateway, VNet peering service, VPN Gateway, Azure Bastion, and Azure DNS.

Services for application protection

These networking resources will help secure and protect your applications or systems in Azure. For example, you can implement networking services like Azure Load Balancer, Firewall, VNet Endpoints, Private Link, and DDoS protection.

Services for application delivery

These services are ideal in use cases involving application delivery. Such networking resources include **Azure Content Delivery Network** (Azure CDN) for content delivery, global web traffic routing using *Azure Front Door Service*, load

balancing your traffic across Azure regions globally using the *Azure Traffic Manager*, and other resources like Application Gateway and Internet Analyzer.

Services for network monitoring

Monitor your network resources using any or a combination of these Azure networking services: Network Watcher, ExpressRoute.monitor, Azure Monitor, and VNet Terminal Access Point (TAP).

We'll explore each of these categories in greater detail throughout the rest of the chapter to gain a better understanding of the different networking services for different use cases.

Azure Networking Services for Connectivity

Azure provides a network infrastructure that is robust, fully managed, and dynamic to support complex network architecture. These network solutions vary from creating public access to application network security and making hybrid connections between infrastructure on-premises and in the cloud.

Azure Virtual Network

Azure Virtual Network (Azure VNet) plays an important role in building networks within the Azure infrastructure. It is a fundamental component of keeping your Azure resources in a private network where you can securely manage and connect to other external networks (public and on premises) over the internet.

Azure VNet goes beyond the usual on-premises and traditional networks. Aside from the benefits of isolation, high availability, and scalability, Azure VNet helps secure your Azure resources by allowing you to administrate, filter, or route network traffic based on your preference.

For example, Azure VMs are connected or attached to an Azure VNet through virtual network interface cards (VNICs), as shown in **Figure 4-1**, where an Azure VM is attached to three NICs: *Default*, *NIC1*, and *NIC2*.

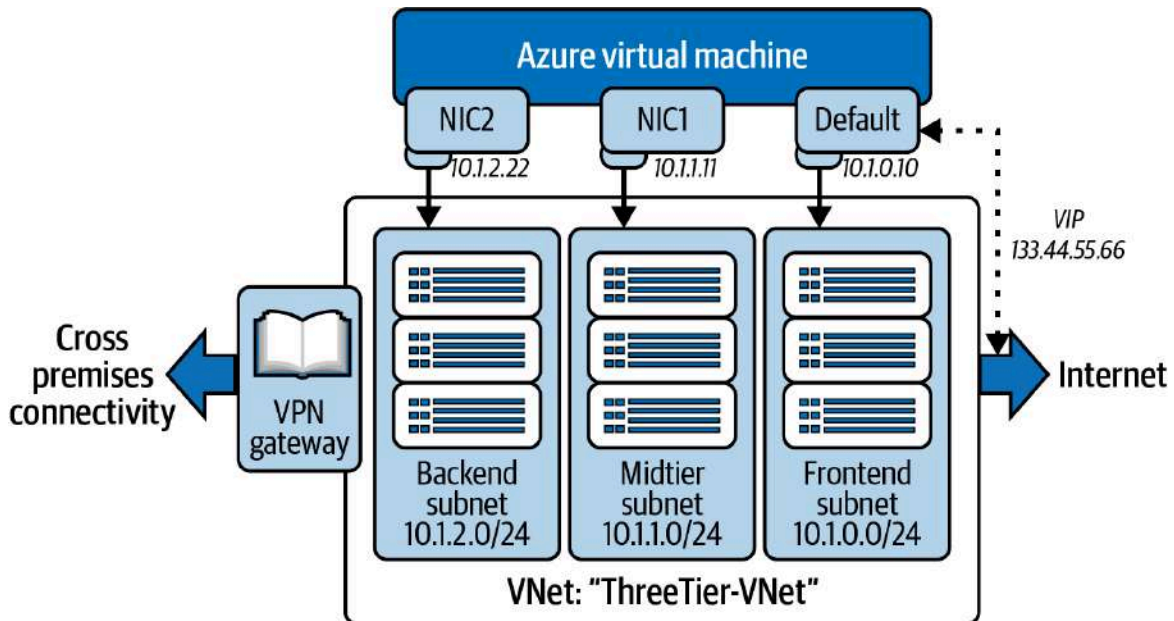


Figure 4-1. ThreeTier-VNet an Azure VM with several NICs

Azure resources need to securely communicate with each other internally on the private network, over the internet, and also networks on on-premises infrastructure. Azure VNet makes this possible. Let's take a closer look at these forms of communication.

Internet communication

Azure VNet by default is capable of communicating outbound to the internet. If you want to communicate to an Azure resource inbound, you may do that using an Azure public IP address or Azure Load Balancer that is set to public.

Communication and connection with Azure resources

Azure resources can securely communicate through the virtual network and VNet peering and by extending virtual network service endpoints. For example, it is possible to

deploy some dedicated Azure resources like **AKS**, Azure Batch, Azure SQL Manage Instance, Microsoft Entra Domain Services, Azure Container Instance (ACI), Azure Functions, Azure App Service Environments, and **Azure VM Scale Sets** within the same virtual network.

Network traffic routing and filtering

The network traffic can be filtered between subnets in several ways. **Network security groups (NSG)** and application security groups can be used to control security rules (inbound and outbound) to control and filter network traffic. Another good option is to use a network VM where you can configure your firewall settings and network rules and optimize your WAN. On Azure Marketplace, you will find some networking **appliance managers** available for external services or within Microsoft.

Azure VNet Peering

Azure Virtual Network peering (VNet peering) allows you to connect several **virtual networks** in Azure. The connectivity and traffic between the VMs in the peered virtual networks uses Microsoft's infrastructure on a secure private network. Virtual networks that are peered can directly share and connect with the resources located in either virtual network.

Currently, Azure supports both VNet peering and global VNet peering. The difference between these two is that global VNet peering connects virtual networks across Azure regions while VNet peering connects virtual networks within the same Azure region.

Figure 4-2 illustrates how VNet peering works between two virtual networks, *VNet A* and *VNet B*, that have several connections to other networking resources.

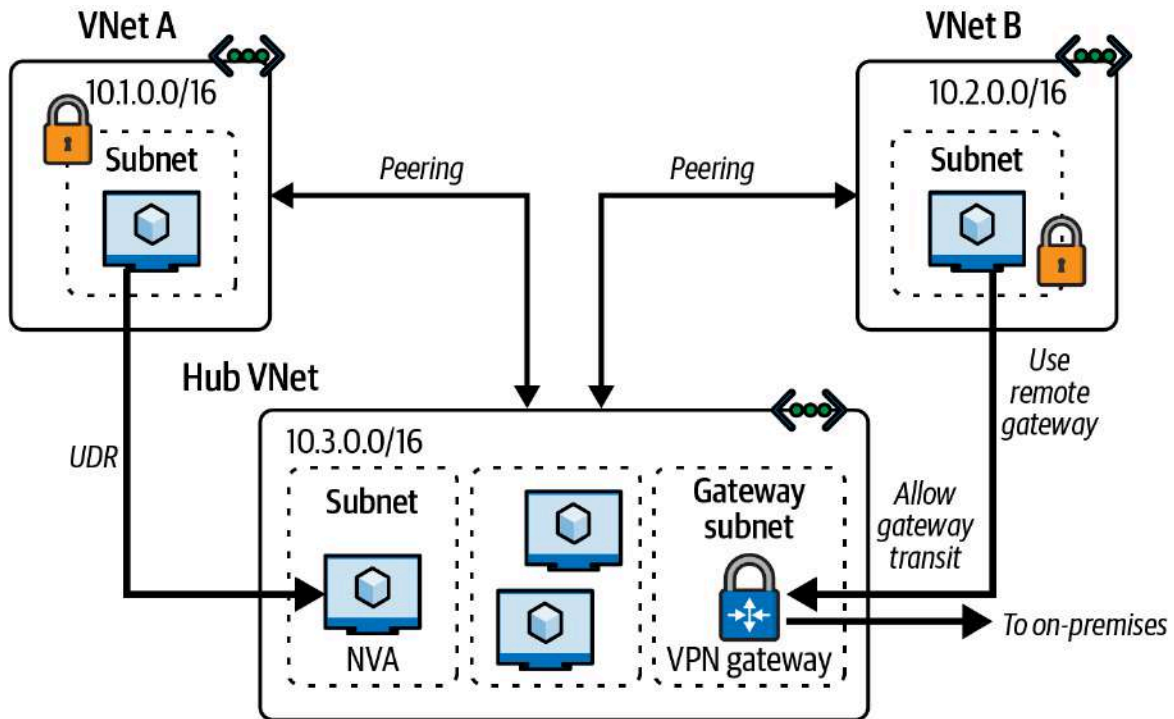


Figure 4-2. VNet peering in Azure

Both VNet peering and global VNet peering support gateway transit, a property in peering that allows a virtual network to use the peered virtual network's *VPN gateway* for cross-premises connectivity or VNet-to-VNet connectivity. You can configure **VPN gateway transit for VNet peering** through Azure Portal, Powershell, ARM template, and Azure CLI.

WARNING

Azure VNet peering issues can occur; therefore, troubleshooting is necessary to understand the root cause of the issue. One of the common troubleshooting questions to consider is whether the virtual networks are in the same subscription or not and if they are in the same region or a different region. For more details about different types of problems with VNet peering, please refer to [Microsoft's documentation on troubleshooting VNet peering issues](#).

On a final note, VNet peering uses IP addresses. There are two types of IP addresses that are used in Azure: public IP and private IP. The private IP addresses are used for the connectivity and Azure resource communication

within the same resource group. On the other hand, public IP addresses are used to allow internet resources to communicate inbound to Azure resources. Using this type of IP address enables communication of Azure resources and public-facing Azure services over the internet.

Learn more in the Microsoft documentation about [IP services](#) and some recommended [best practices for Azure Virtual Network](#).

Azure Virtual Wide Area Network

Azure Virtual WAN is a managed networking service and unified framework for networking, security, and routing features. The [Azure global network](#) enables Azure Virtual WAN to be available. It includes site-to-site, point-to-site VPN connectivity, ExpressRoute, etc.

Virtual WAN helps organizations or business units to connect to the internet and other Azure resources, for example, [networking and remote user connectivity](#), which is effective and useful for those who prefer to work from home or other remote locations. It can also be used to move existing infrastructure or data center from on premises to Microsoft Azure with the help of utilizing [Azure Virtual WAN](#).

Features of Azure Virtual WAN include:

- Branch connectivity (via connectivity automation from Virtual WAN partner devices such as SD-WAN or VPN CPE)
- Site-to-site VPN connectivity
- Point-to-site VPN connectivity for remote users
- Private connectivity using ExpressRoute
- Intracloud connectivity for virtual networks
- Interconnectivity using VPN ExpressRoute
- Routing, Azure Firewall, and encryption for private connectivity

Azure Virtual WAN also has benefits like integrated connectivity solutions in the hub and spoke, automated spoke configuration, and troubleshooting and monitoring tools. When configuring it, you need Virtual WAN resources such as Virtual Hub, Hub VNet Connection, hub-to-hub connection, a hub route table, and site resources.

Figure 4-3 shows an example of Azure Virtual WAN being used and implemented for remote connectivity.

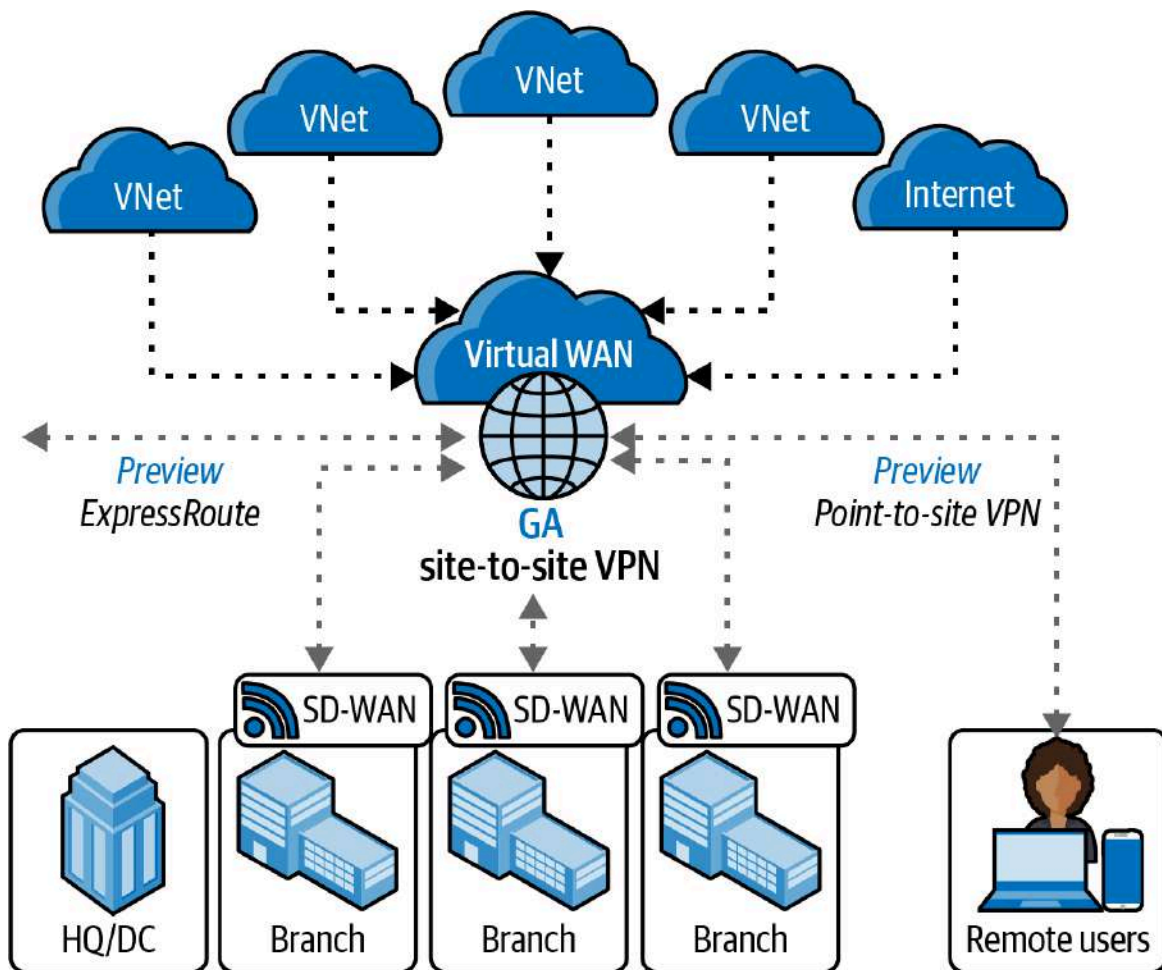


Figure 4-3. Azure Virtual WAN (adapted from an image by [Microsoft documentation](#))

TIP

Each migration is unique; use the options and resources available to help you navigate your [migration to Azure Virtual WAN](#).

Azure Virtual WAN is a broad and complex topic. For a deeper dive, see [Microsoft's Virtual WAN documentation](#).

Azure ExpressRoute

Azure ExpressRoute allows you to expand networks that are on premises into Microsoft's cloud infrastructure using a connectivity provider over a private connection. Basically, this networking service enables connecting your on-premises networks with Azure. This connection between an on-premises network and Azure can be initiated using an any-to-any (IPVPN) network with Layer 3 connectivity, which enables you to interconnect with Azure to your own on-premises WAN or data center.

The connection in Azure ExpressRoute is private, and the traffic with this connection does not go over the internet. This means that the connections made using ExpressRoute promise speed, reliability, **availability**, and better security compared to connections over public networks.

A secure network connection can be established with other cloud resources in Microsoft like Microsoft 365, Dynamics 365, and Microsoft Azure, as shown in [Figure 4-4](#).

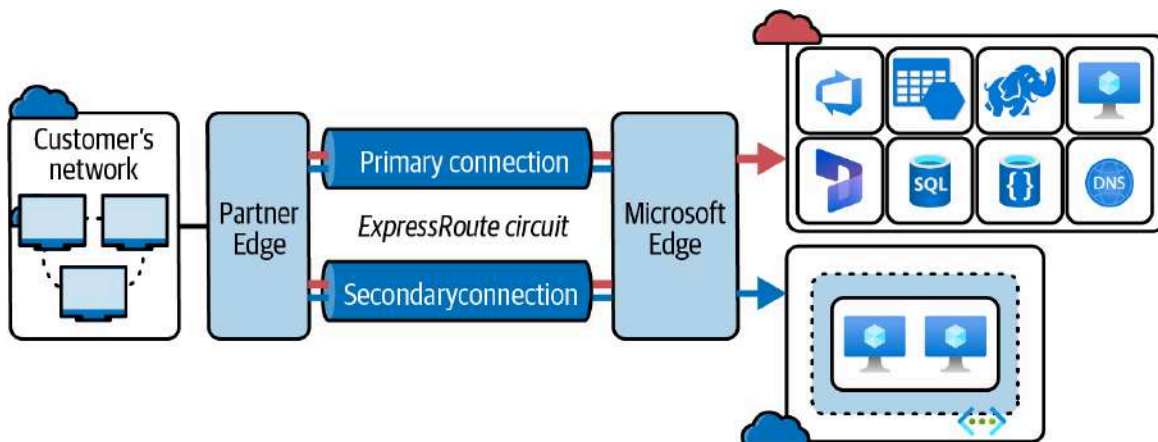


Figure 4-4. Azure ExpressRoute connections between public and on-premises networks (adapted from an image by [Microsoft documentation](#))

ExpressRoute has several useful features such as support for different **connectivity models** between your on-premises network and Azure. As shown in [Figure 4-5](#), these connectivity models can be implemented using

service providers or directly. Service providers currently use three types of models: *cloud exchange co-location*, *point-to-point Ethernet connection*, and *any-to-any (IPVPN) connection*. ExpressRoute Direct can also be used for direct connection to Microsoft Azure.

ExpressRoute Direct is a good Azure service if you want to directly connect your network into Microsoft's network globally at peering locations at up to 10 GBps or 100 GBps connectivity. You can **create an ExpressRoute Direct connection** through the Azure Portal, Azure CLI, and Azure PowerShell. Learn more about **ExpressRoute Direct circuits**, workflows, VLAN tagging, SLA, and pricing in the Microsoft documentation. Azure Storage can be integrated with ExpressRoute Direct if you have a use case that requires ingestion of data.

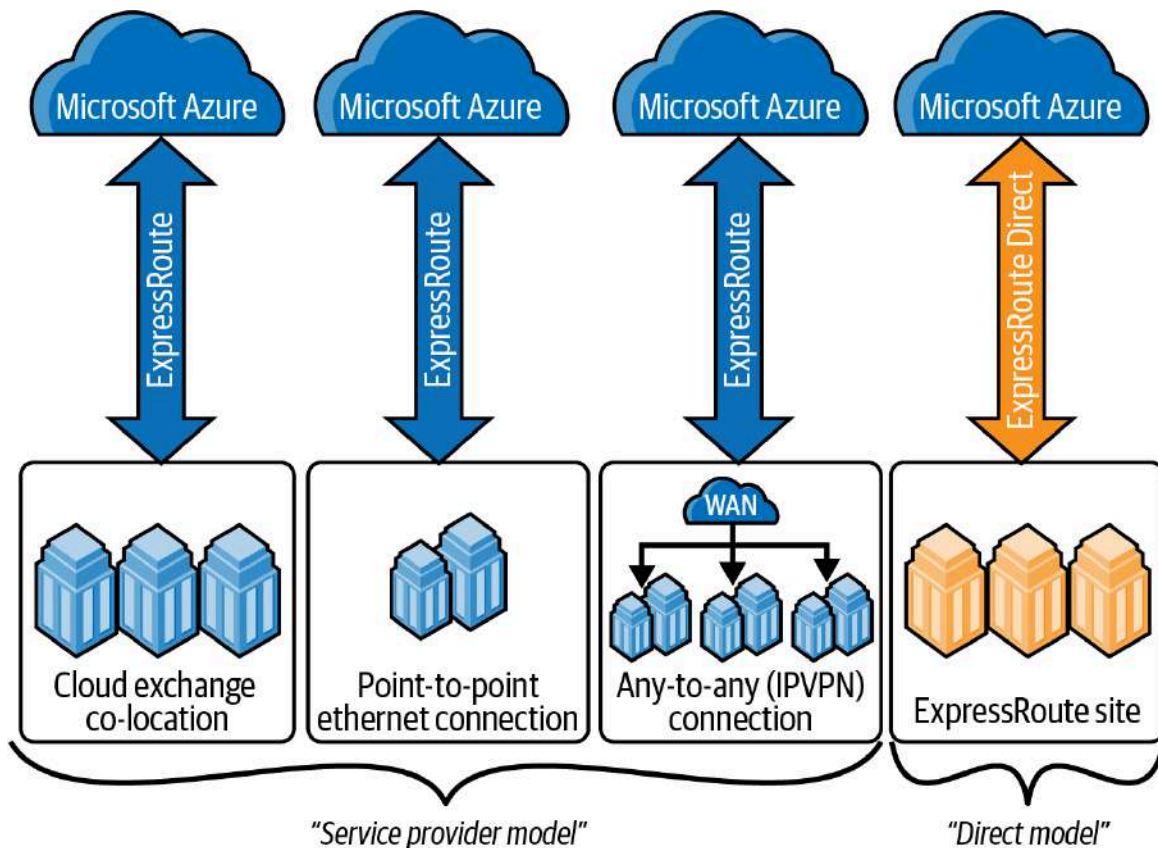


Figure 4-5. Different connectivity models of Azure ExpressRoute

WARNING

When you configure and manage ExpressRoute, **network address translations (NATs)** are needed to connect to Microsoft cloud services. Connectivity service providers usually offer NAT management as a service. Otherwise, there are some **NAT requirements for Microsoft peering and Azure public peering**. There are also some **ExpressRoute routing** requirements that you need to be aware of.

Azure ExpressRoute has the option to implement private peering for Azure resources like VMs and **Azure Virtual Desktop RDP Shortpath** within the virtual networks.

Azure ExpressRoute Global Reach

In addition to the different connectivity models of Azure ExpressRoute for direct connection and service providers, there is another related networking service called ExpressRoute Global Reach, which is designed for scenarios like connecting an organization's branch offices across the world.

For example, if your ExpressRoute service provider does not operate in certain locations or countries of your office branch, you may consider using the service provider locally in that location or country. Microsoft will be able to help connect your organization's branch offices to the ExpressRoute service provider in your main office location using its global network.

The locations of ExpressRoute are the starting connection point to the **globally distributed network infrastructure of Microsoft**. Due to this capability ExpressRoute users are able to connect around the world using Microsoft's network.

Azure VPN gateway

A VPN gateway is a specific type of virtual network gateway used to send encrypted traffic between an Azure virtual network and an on-premises location over the public internet. You can also use a VPN gateway to send encrypted traffic between Azure virtual networks over the Microsoft network. Each virtual network can have only one VPN gateway. However,

you can create multiple connections to the same VPN gateway. When you create multiple connections to the same VPN gateway, all VPN tunnels share the available gateway bandwidth.

Different types of VPN gateway connections

To create a VPN gateway connection, we need to know the different configurations. These different types of connections will give you a clear overview and understanding on what is ideal based on what suits your business requirements.

Site-to-site VPN (S2S)

The site-to-site (S2S) connection runs over the IPsec/IKE ((IKEv1 or IKEv2) VPN tunnel. IPsec VPN is one of the common VPN protocols used for establishing a VPN connection. The S2S VPN gateway connections can be hybrid or cross-premises use cases. This type of gateway also requires a **VPN device** that must be located on premises. This VPN device should have a public IP address designated to it.

Point-to-site VPN

The point-to-site (P2S) type of VPN connection is applicable if you want to connect a client computer to a virtual network like Azure VNet. This type of VPN connection needs to be initiated from the client computer, which is useful for remote work.

VNet-to-VNet (IPsec/IKE VPN tunnel)

Unlike the P2S VPN connection, the VNet-to-VNet connection is for connecting between VNets, even virtual networks on premises. This type of VPN gateway uses a secure connection through IPsec/IKE. Configuring for multiple site connections is also possible using this type of VNet-to-VNet connection.

Site-to-site and ExpressRoute

The network traffic for the S2S VPN gateway is securely encrypted through the public internet. ExpressRoute also creates a private and direct connection from a WAN to Azure and other Microsoft services. This means that configuring ExpressRoute together with S2S VPN has advantages, especially if it is set for the same virtual network.

NOTE

The VNets you connect using VNet-to-VNet (IpSec/IKE VPN tunnel) connections can be in different or the same Azure regions, subscriptions, and deployment models.

One of the important steps when configuring the VPN gateway is choosing the appropriate type. A virtual network can have only one VNet gateway. For example, if you are configuring VNet gateway for VPN, then you would need to use `-GatewayType Vpn` and for ExpressRoute you would similarly use its keyword value. Check the guide for [configuring VPN gateway settings](#) and some requirements you need to know.

Azure NAT gateway for virtual networks

Azure Virtual Network NAT helps in simplifying the outbound internet connectivity for virtual networks (one or several subnets). Once a subnet is associated with a NAT gateway, the NAT gives the benefit of providing a source network address translation (SNAT). For example, you can utilize Azure NAT Gateway to securely connect to your VMs using dedicated public IP while having control over the ports you share. When a VNet NAT is configured on a subnet level, the outbound connectivity will use the specified static public IP addresses. [Figure 4-6](#) illustrates how Azure VNet NAT gateway works with a VM with public IP. The Azure VMs in Subnet A have instance-level Public IP addresses but the VMs on Subnet B do not.

The network traffic inbound are directed to Subnet A virtual machines that are still directed to an instance-level IP. However, all the outbound traffic from both Subnets A and B is routed through Azure NAT Gateway.

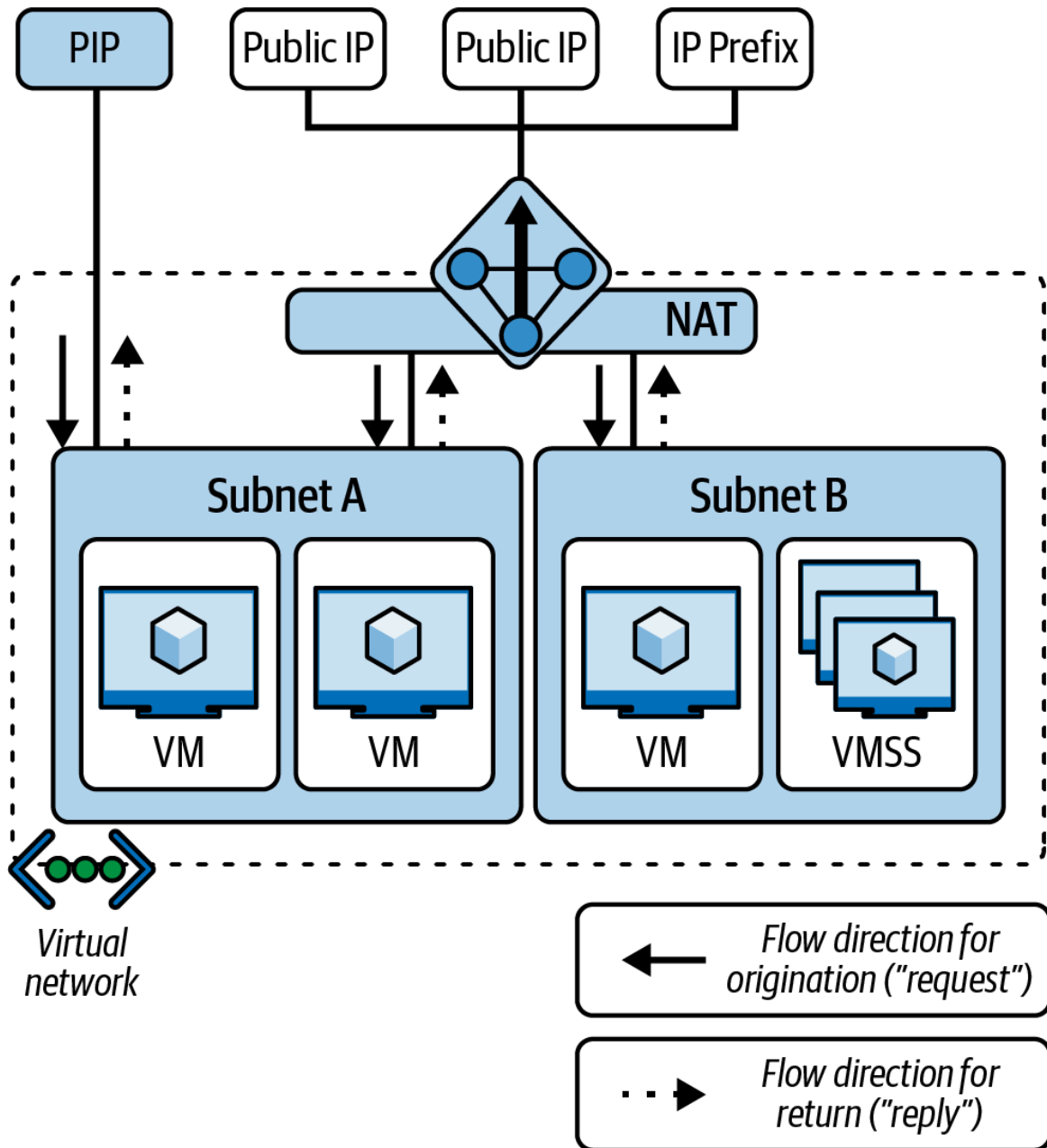


Figure 4-6. Instance-level Public IP with a VM and NAT

The following are some of the benefits of using Azure VNet NAT:

Security and privacy

Azure VNet NAT gateway doesn't need public IP addresses; therefore, the connection between virtual networks is fully private and secured. Resources can still connect to external resources outside the VNet even without public IP addresses.

Scalability

All compute resources belong to a subnet. All subnets can communicate and use the same resource in the same virtual network. Automatic scaling is possible with the use of a public IP prefix, which assists in identifying and scaling how many outbound IP addresses are required.

Resiliency

NAT does not have any individual dependency on other compute instances, which is the result of it being a distributed and fully managed service. This feature makes it very resilient as a software-defined networking service.

Performance

The **performance of a NAT gateway** is satisfactory because it is a software defined networking service; when it is running it will not have a negative effect on network bandwidth.

Azure VNet NAT gateway provides the ability to control who has access to your organization's resources and what locations they can be accessed from. For these reasons, NAT gateway can be useful for whitelisting a group of people working as contractors for a company.

NAT is not applicable or supported to work with basic public IP addresses and load balancers. If you need to use NAT gateway in your implementation, you would need to use standard versions virtual networks (one or several subnets). Once a subnet is associated with a NAT gateway, the NAT gives the instead or **upgrade Azure Public Load Balancers to higher versions**.

If you want to try designing a virtual network using NAT gateway, see [the Microsoft documentation](#) and a video about it from [Azure Friday](#).

Azure Domain Name System

Azure Domain Name System (DNS) is a hosting service for DNS domains that provides name resolution using the Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services. You can't use Azure DNS to buy a domain name. For an annual fee, you can buy a domain name by using App Service domains or a third-party domain name registrar. Your domains then can be hosted in Azure DNS for records management.

The following are some of the key features of Azure DNS:

Reliability and performance

DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers. Azure DNS uses anycast networking. Each DNS query is answered by the closest available DNS server to provide fast performance and high availability for your domain.

Security

Azure DNS is secured because it is linked to Azure Resource Manager, which is associated with user identity control services like Azure RBAC and [Azure Resource Lock](#). Resource locking is used to prevent other users in your organization from accidentally deleting or modifying critical resources.

Alias records

Azure DNS supports alias record sets. You can use an alias record set to refer to an Azure resource, such as an Azure public IP address, an Azure Traffic Manager profile, or an Azure Content Delivery Network (CDN) endpoint. If the IP

address of the underlying resource changes, the alias record set seamlessly updates itself during DNS resolution. The alias record set points to the service instance, and the service instance is associated with an IP address. Also, you can now point your apex or naked domain to a Traffic Manager profile or CDN endpoint using an alias record.

WARNING

Domain Name System Security Extensions (DNSSEC) are currently not supported in Azure DNS. However, a work-around alternative is using HTTP/TLS in the configuration of your applications. If your DNS zones require DNSSEC, host them externally through DNS providers.

By using Azure DNS, you host your own domain websites in Azure, manage your DNS records and integrate them with the resources hosted in Azure. If you have an existing domain, there are a number of ways to host it in Azure DNS. Microsoft has different guides for setting up your **own custom domain** on a function app, web app, blob storage, or other Azure resource.

Azure Bastion

The Azure Bastion service is a new, fully platform-managed PaaS service that enables secure VM connections without exposing the confidential ports of your network to the public Internet. It provides secure and seamless RDP/SSH connectivity to your VMs directly in the Azure Portal over TLS. When you connect via Azure Bastion, your VMs do not need a public IP address.

You can connect securely via RDP and SSH to all of the VMs in the virtual network where Azure Bastion is provisioned. By doing this, you are securely connecting to your VMs and are not exposing RDP/SSH ports to the public internet.

Key benefits and uses of Azure Bastion include:

RDP and SSH directly in a web browser

Since Azure Bastion uses an HTML5-based web client, you can access your VMs on any device. This means you don't need to download a supported RDP or SSH client to connect to a VM. You can connect to your VMs securely using RDP and SSH sessions directly on any web browsers through the Azure Portal.

Public IP not required on the Azure VM

Azure Bastion opens the RDP/SSH connection to your Azure VM using a private IP on your VM. You don't need a public IP on your VM.

Save time managing network security groups (NSGs)

Azure Bastion is hardened internally to provide you secure RDP/SSH connectivity. You don't need to apply any NSGs to the Azure Bastion subnet. Because Azure Bastion connects to your VMs over a private IP, you can configure your NSGs to allow RDP/SSH from Azure Bastion only. This removes the hassle of managing NSGs each time you need to securely connect to your VMs.

Protection against port scanning

VMs are secured and protected against port scanning by rogue and malicious users located outside your virtual network.

Protect against zero-day exploits

Azure Bastion protects against zero-day exploits. A zero-day exploit is a cybersecurity attack that takes advantage of a security vulnerability. The term is also known as "0-day," which means that the developer had 0 days to work on an

update to fix the issue. Using anti-virus, firewall, and data protection tools can help avoid these security threats. You'll learn more about zero-day exploits and cloud security on Azure's cloud resources in [Chapter 9](#).

Azure Bastion is a flexible and secure way to connect VMs on the go. To learn more, check out this video guide from [Azure Friday](#).

Services for Application Protection

This section describes networking services in Azure that help protect your network resources. Using any or a combination of these networking services in Azure, you can secure your applications and workloads in the cloud.

Azure Firewall

Azure Firewall is a cloud-native intelligent network firewall security service that provides threat protection for your cloud workloads running in Azure. It's a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability.

Following are the different types of Azure Firewall to help you identify which is right for your needs:

Azure Firewall Standard

Azure Firewall Standard provides [Layer 3 to Layer 7 firewall filtering](#) and threat intelligence feeds from Microsoft Cyber Security. These threat feeds can be used to notify about any important alert and can even deny network traffic from/to any malicious domains and IP addresses to protect Azure resources against possible hacking or attacks.

Azure Firewall Premium

Azure Firewall Premium provides advanced capabilities such as signature-based IDPS that enable detection of attacks

by analyzing and detecting specific patterns, such as byte sequences in network traffic or any known malicious instruction sequences used by malware. This tier also supports third-party offerings available from WatchGuard, Sophos, Palo Alto, Check Point, and the like.

WARNING

There are documented, **known issues for both the standard and premium version** of Azure Firewall. Note that the Azure Firewall tier names might change, so it is important to keep up with the recent updates on this security service on Azure.

Aside from the different Azure Firewall categories and tiers, you can centrally manage Azure Firewalls across multiple Azure subscriptions using Azure Firewall Manager. It supports setting up centralized security and route management. This firewall policy can be utilized by applying a common set of firewall rules in your network or application in your Azure tenant. Azure Firewall Manager supports firewalls in environments like VNets and **Virtual WANs (Secured Virtual Hub)**.

Azure Firewall is an excellent choice to secure your Azure resources. The team behind Azure Firewall will continue to add features to both the standard and premium tiers.

Azure DDoS Protection

Azure DDoS Protection provides countermeasures against the most sophisticated **DDoS (distributed denial of service) threats** that can cause severe and highly impactful loss. It provides advanced DDoS mitigation features for your applications and resources.

Customers using Azure DDoS Protection also have access to support and communicate with DDoS experts using **Azure DDoS Rapid Response** during an active attack. An Azure DDoS Protection standard plan is a prerequisite to this support.

Figure 4-7 illustrates how Azure DDoS Protection works in securing an application gateway in a virtual network.

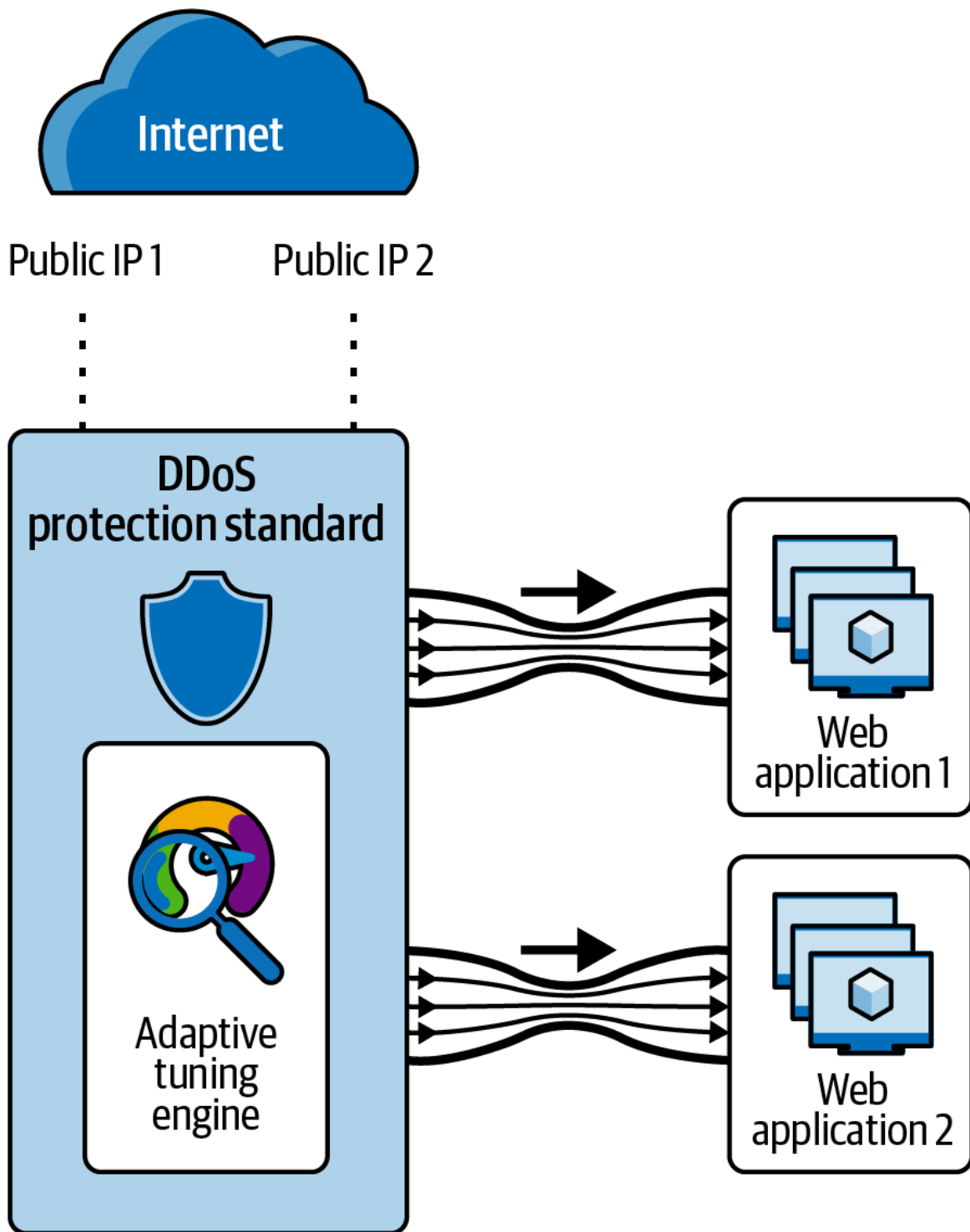


Figure 4-7. Azure DDoS Protection for a web application in a virtual network

Engineers with a role involving securing resources in Azure should learn the basic concepts of Azure security and protecting Azure resources from attacks. Check out Microsoft's learning path for [for an introduction to Azure DDoS Protection](#).

Azure Private Link

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure-hosted customer-owned/partner services over a private endpoint in your virtual network. Traffic between your virtual network and the service travels through the Microsoft backbone network. Exposing your service to the public internet is no longer necessary. You can create your own private link service in your virtual network and deliver it to your customers.

Web Application Firewall

Azure Web Application Firewall (WAF) provides protection to your web applications from common web exploits and vulnerabilities such as SQL injection and cross-site scripting. [Figure 4-8](#) illustrates how Azure WAF works.

Azure WAF provides out-of-the-box protection from the Open Worldwide Application Security Project's top 10 vulnerabilities via managed rules. Additionally users of WAF can also configure custom rules, which are customer-managed rules to provide additional protection based on source IP range, and request attributes such as headers, cookies, form data fields, or query string parameters.

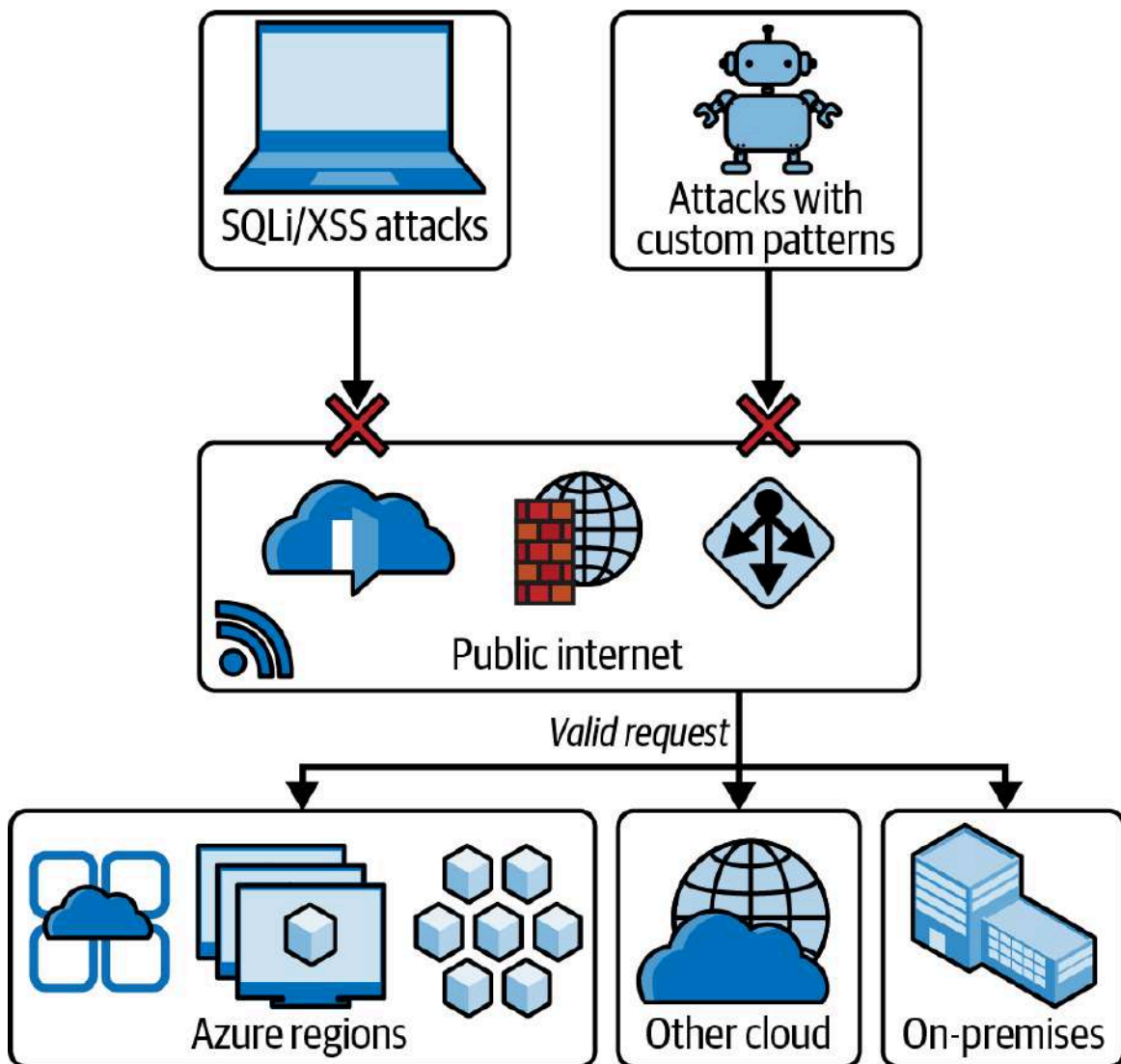


Figure 4-8. Azure Web Application Firewall

Users or Azure administrators can choose to deploy Azure WAF with Application Gateway, which provides regional protection to entities in public and private address spaces. It is also possible to deploy Azure WAF with Azure Front Door, which provides protection at the network edge to public endpoints.

Network security group (NSG)

Network security groups (NSGs) are built-in tools for network control that allow us to control incoming and outgoing traffic on a network interface or at the subnet level. They contain sets of rules that allow or deny specific

traffic to specific resources or subnets in Azure. An NSG can be associated with either a subnet (by applying security rules to all resources associated with the subnet) or a network interface card (NIC), which is done by applying security rules to the VM associated with the NIC.

Figure 4-9 illustrates how NSGs are used at different levels to protect several VMs in different subnets in a virtual network.

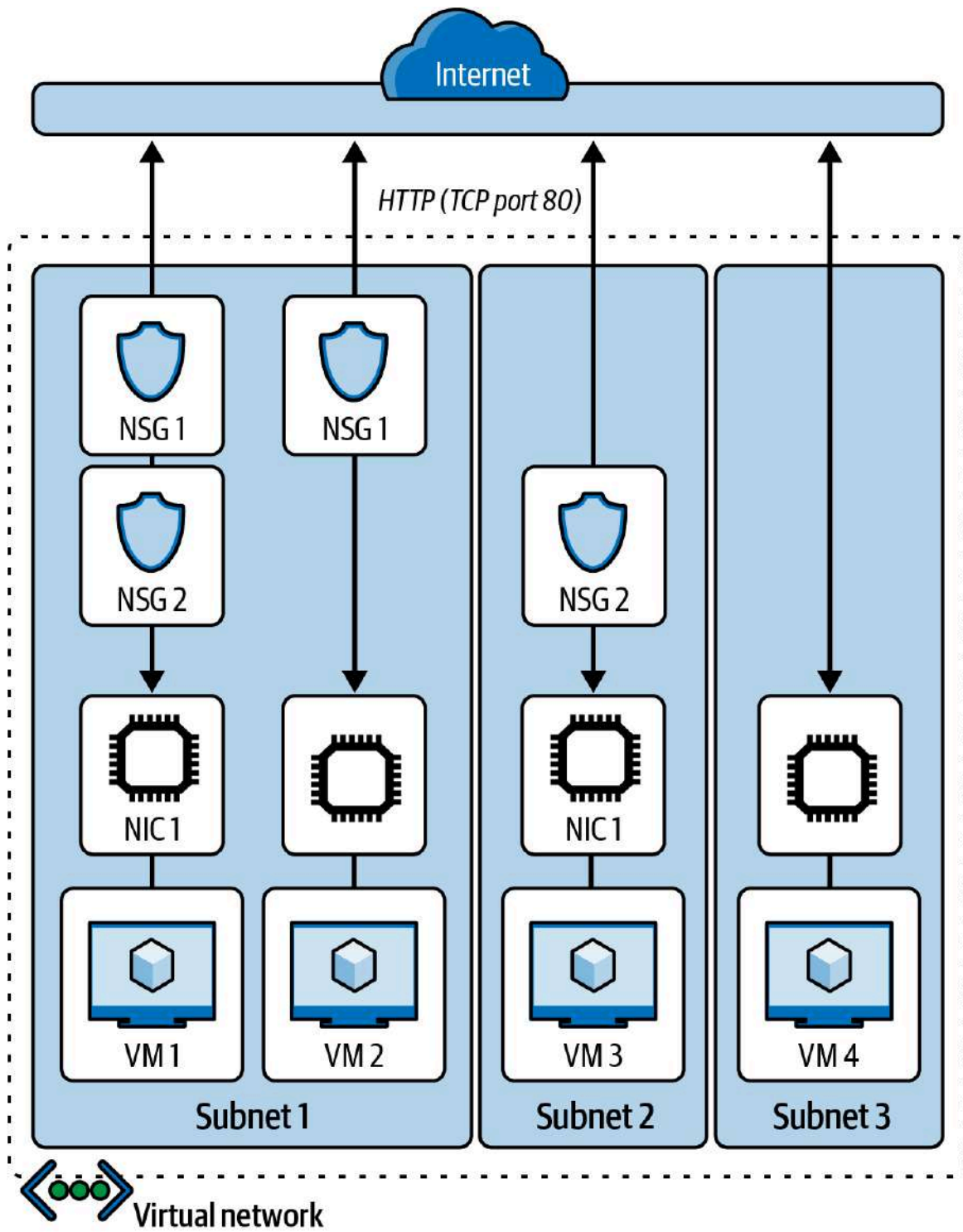


Figure 4-9. A virtual network protected with NSGs

Azure Load Balancer

Load balancing is a term commonly used for workload distribution across several computing resources. This process helps optimize the uses of resources, fix issues with response times, increase throughput, and prevent single resource overloading. Typically, load balancers can be public or internal. Public load balancer supports network traffic for your virtual machines by translating your private IP addresses to public IP addresses of your VMs. On the other hand, internal load balancers support load balancing inside your virtual network. If you have a hybrid infrastructure as a setup, this type of private load balancing is also possible with a frontend load balancer. In Azure, services for load balancing include *Azure Load Balancer*, *Traffic Manager*, *Azure Front Door*, and *Application Gateway*.

Azure Load Balancer is a load-balancing networking service that distributes traffic across multiple VMs (Azure VMs) or a group of resources in a single *Azure region*. This resource is best known for its performance with ultra-low latency. The **algorithm for Azure Load Balancer** uses a tuple-based hashing distribution, which means that the tuple hashes itself on the bases of its elements.

Figure 4-10 shows a 5-tuple hash (source IP, source port, destination IP, destination port, and protocol type) that is mapped to the traffic to any available servers.

The hash-based and source IP affinity are both supported in the Azure Load Balancer service. This means you have the option to set up the configuration of your preferred **distribution mode** for traffic distribution.

Key uses of Azure Load Balancer

Azure Load Balancer is a powerful resource for creating applications that are highly available and scalable for inbound and outbound connections.

Uses and benefits of using the standard Azure Load Balancer include:

- Improve the distribution of Azure resources with better availability

- Configure connectivity and load balancing of Azure VMs for incoming and outgoing traffic
- Monitor Azure resources that are being load-balanced and distributed by the Azure Load Balancer
- Require **port forwarding**, load-balance with IPv6, multiple ports or IP addresses
- Ability to migrate load-balancing resources across different Azure regions

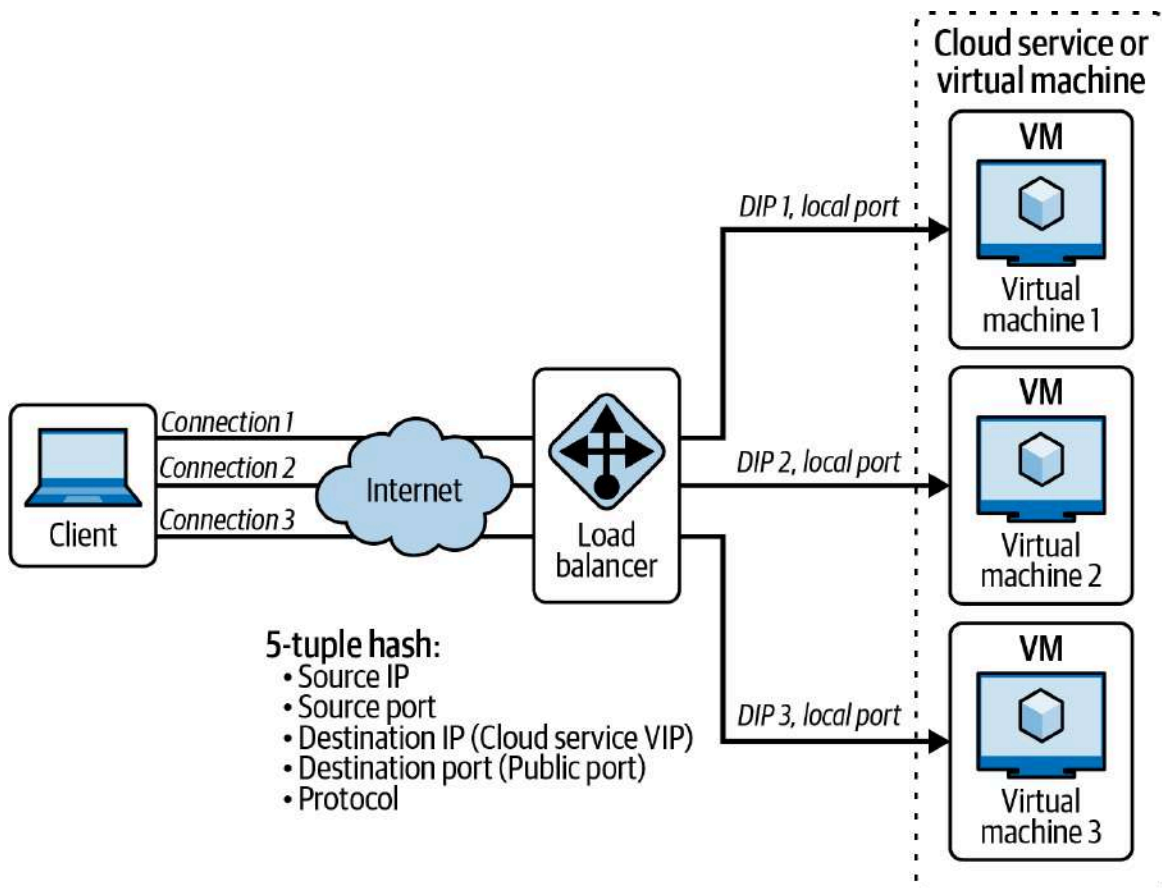


Figure 4-10. Azure Load Balancer with 5-tuple hash distribution

The load balancer operates in the *Open Systems Interconnection* (OSI) model at Layer 4, which distributes traffic based on protocol, source IP address and port numbers, and destination IP address and port. The main purpose of the load balancer is to distribute the inbound flows from both backend and frontend pool instances.

The Azure Load Balancer components are described in **Table 4-1**; a few key components can be configured in your Azure subscription on the Azure Portal. You also have the option to configure using other Azure tools like Azure PowerShell, Azure CLI, or ARM templates.

Table 4-1. Azure Load Balancer Components

Component	Description
Frontend IP configuration	The IP address (public or private) you set on a load balancer will be your client's access point. The type of load balancer depends on what kind of IP address you have. For example, the private IP address is for the internal load balancer while the public IP is for the public load balancer.
Backend pool	It is generally recommended to add more instances to the backend pool to cost-effectively scale and handle the high demands of incoming traffic. To optimize operations, it is worth considering designing for the least number of individual backend pool resources.
Load balancer rules	A load balancer configured with protocol <i>-all</i> and port <i>-0</i> is known as following the high availability (HA) port rule. This rule enables a single rule to load-balance all TCP and UDP flows that arrive on all ports of an internal Standard Load Balancer. The HA port's load-balancing rules help with critical scenarios, such as high availability and scale for network virtual appliances (NVAs) inside virtual networks.
Health probes	A health probe is used to determine the health status of the instances in the backend pool. During load balancer creation, configure a health probe for the load balancer to use. This health probe will determine if an instance is healthy and can receive traffic.

Component	Description
Inbound NAT rules	An inbound NAT rule forwards incoming traffic sent to a frontend IP address and port combination. The traffic is sent to a specific VM or instance in the backend pool. Port forwarding is done by the same hash-based distribution as load balancing.
Outbound rules	An outbound rule configures outbound NAT for all VMs or instances identified by the backend pool. This rule enables instances in the backend to communicate (outbound) to the internet or other endpoints.
High availability ports	The HA ports load-balancing rules help you with critical scenarios, such as high availability and scale for NVAs inside virtual networks. The feature can help when a large number of ports must be load-balanced.

TIP

If you need to set up load balancers in more than one frontend, you can also configure it for multiple frontends. Check out Microsoft's guide for [multiple frontends for Azure Load Balancer](#)

To learn more about which load-balancing network service is suitable for specific scenarios, [check out this discussion](#) about reviewing network options and identifying requirements for workload networking.

There are several load-balancing options in Azure; the load-balancing decision tree guide provided by Microsoft is shown in [Figure 4-11](#). You can also read the details of this flowchart in Microsoft's guide for [load-balancing options](#).

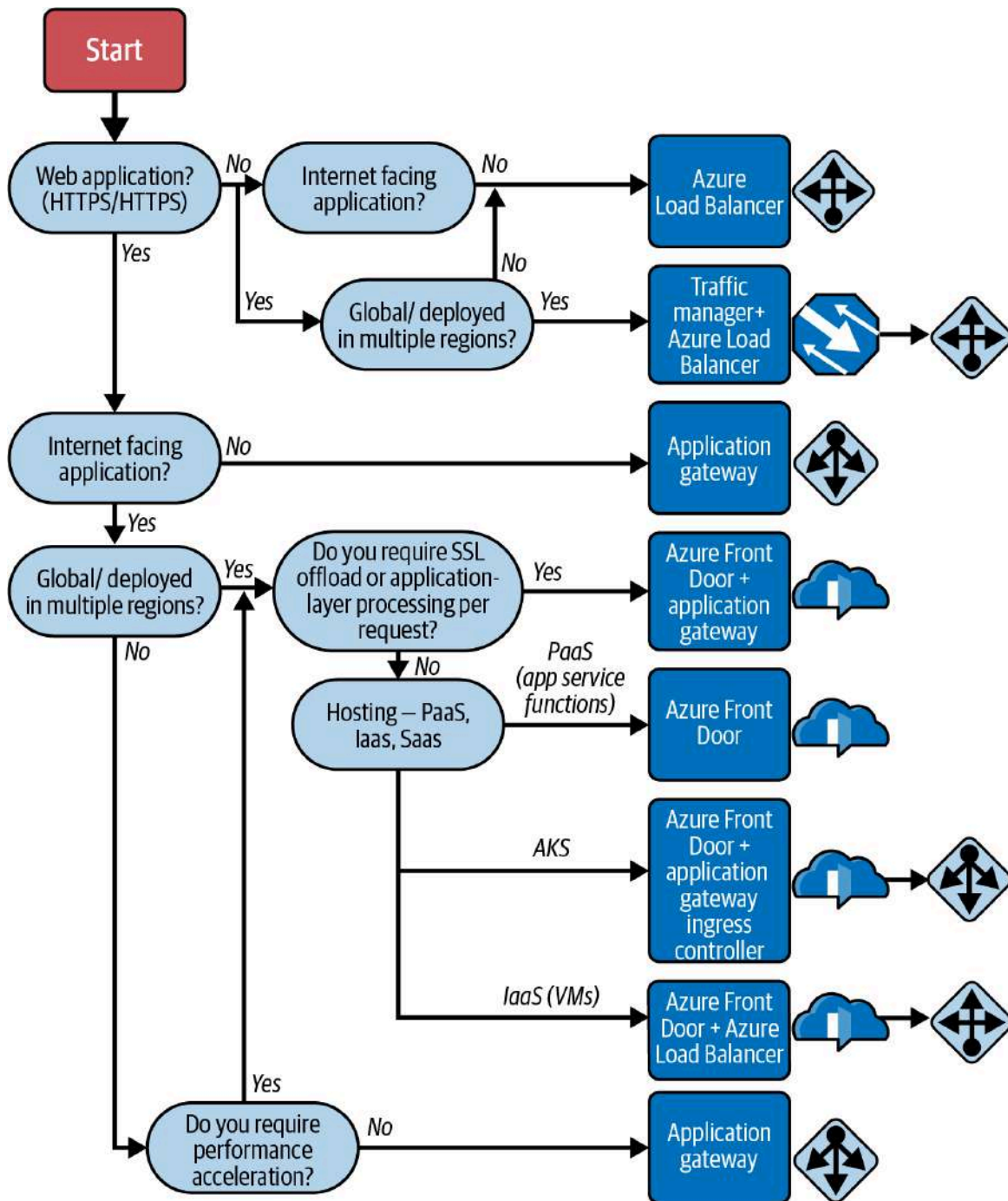


Figure 4-11. Microsoft's decision tree guide for load-balancing options (adapted from an image by *Microsoft documentation*)

Each application can have different implementations and can consist of multiple workloads. Each workload needs to be evaluated separately.

Azure Networking Services for Application Delivery

Application delivery is another category of Azure networking services. Azure networking services like Azure Front Door, Azure CDN, Azure Balancer, etc., will help speed up delivery and user experience on your cloud-hosted applications globally.

Azure Front Door

Azure Front Door is a global, scalable entry point that uses the Microsoft global edge network to create fast, secure, and widely scalable web applications. With Front Door, you can transform your global consumer and enterprise applications into robust, high-performing personalized modern applications with content that reaches a global audience through Azure.

With the Azure Front Door service, the content on your applications will be delivered quickly to your application users wherever they are located. This service uses **Microsoft's global edge network** that interconnects to many points of presence (POPs) distributed around the world.

With Front Door you can develop, operate, and scale your dynamic web applications and even static content. It also allows you to configure, manage, and monitor the global routing for your web traffic by optimizing for top-tier end-user performance and reliability through quick global failover.

Following are additional key benefits and uses of Azure Front Door:

- Cookie-based session affinity, which is useful for keeping a user's session active on the same device or server
- **SSL offloading** and certificate management
- Custom domains if you want to configure your own domains
- Integration with **Web Application Firewall (WAF)** for security
- HTTP/HTTPS redirection helps ensure that the connection between your web client and server are secured and encrypted

- URL rewrites and custom redirection
- Smart monitoring for resources in the backend that will help track and debug issues
- Multiple website hosting and support for **wildcard domains**
- End-to-end IPv6 connectivity and HTTP/2 protocol support

There are different tiers to choose from: Azure Front Door Standard and Azure Front Door Premium. For more details about the different tiers, see Microsoft's guide for **Azure Front Door tiers**.

Azure Application Gateway

Azure Application Gateway is a load balancer for web traffic. It enables you to manage and control the traffic to your web applications. It is an **application delivery controller (ADC)** as a service that supports several Layer 7 load-balancing capabilities for your applications.

Microsoft's Azure Application Gateway is ranked #3 in application delivery controller solutions based on a **Peerspot.com ranking survey**.

Features of Application Gateway include:

- Support for autoscaling (scale up and scale down) based on the ongoing traffic load of your application
- SSL/TLS termination at the gateway to secure unencrypted traffic to servers
- Zone redundancy and support for multiple availability zones to ensure high availability and fault tolerance
- URL-based routing, multiple site hosting and redirection
- Protection and security using Web Application Firewall (WAF)
- Use as an **ingress controller** for an Azure Kubernetes Service (AKS) cluster

- Integration with Azure Monitor for monitoring, logging, and insights

Azure has several fully managed load-balancing solutions that suit different use cases or needs. Azure Load Balancer is an alternative if you want to do load balancing at the network layer level. There's also Azure Front Door for optimizing global routing of web traffic and Azure Application Gateway if you need server load balancing at the application layer.

Azure Traffic Manager

The Azure Traffic Manager is a DNS-based traffic load balancer. This networking service allows you to effectively distribute traffic to your public-facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness.

Key benefits of Azure Traffic Manager include:

Integrate and associate hybrid applications

Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including “burst-to-cloud,” “migrate-to-cloud,” and “failover-to-cloud” scenarios.

Improve availability, maintainability, and performance

Azure Traffic Manager promises to deliver high availability for critical applications by monitoring endpoints and automatic failover if an endpoint goes down. You can have planned maintenance done on your applications without downtime. Traffic Manager can direct traffic to alternative endpoints while the maintenance is in progress. It also can help improve your application's performance by directing web traffic to the endpoint with the lowest latency.

Manage complex traffic distribution and advanced deployments

By using **Nested Traffic Manager profiles**, you can use multiple traffic-routing methods to create flexible rules to scale to the demands of more complex and larger deployments.

NOTE

Priority, Weighted, Geographic, Performance, MultiValue, and Subnet are the **traffic-routing methods** supported by Azure Traffic Manager. These routing methods are used to check how network traffic is routed to different service endpoints.

Figure 4-12 shows how Azure Traffic Manager can be used to direct a web application's client request to a specific endpoint based on the traffic-routing method being configured.

WARNING

Azure Traffic Manager and Azure Front Door have similarities and differences. For example, for routing options, Traffic Manager uses on-premises routing at the DNS layer, while Azure Front Door works with HTTP requests, which have independent scalability. Traffic Manager works with any protocol like UDP, TCP, HTTP, and more. However, Front Door uses HTTP acceleration, which means that its traffic is on a proxy on Microsoft's edge network.

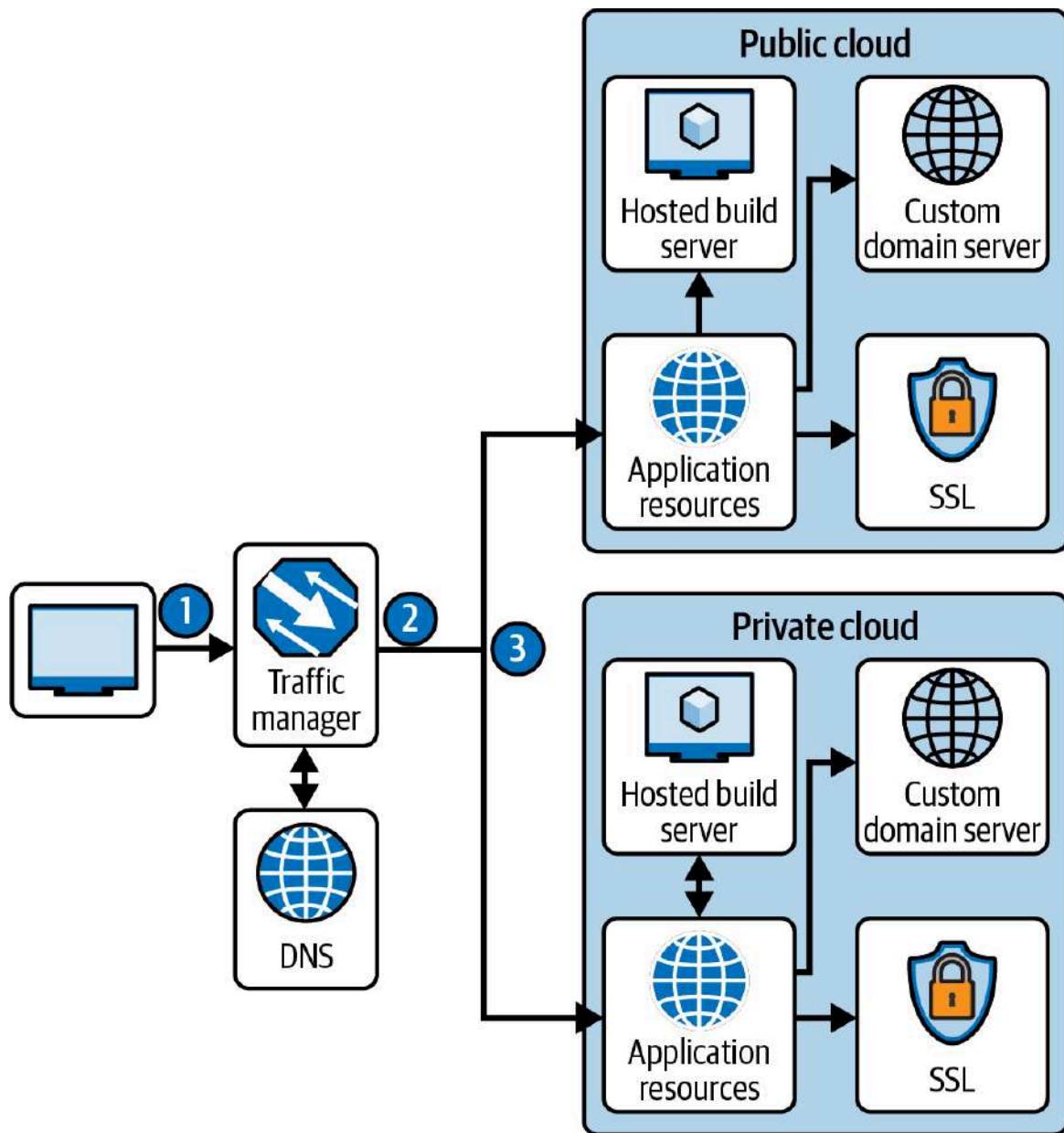


Figure 4-12. An example of Azure Traffic Manager being used for a web application to scale and load-balance traffic in a hybrid cloud environment

If you have several instances in different Azure regions and one of these instances fails during the health check, the traffic for your application is directed to the Azure region that is healthy. However, a performance problem can occur in the latency of the traffic to a region located far away.

Therefore, there are some performance considerations when using and managing Azure Traffic Manager. There are several tools you can use to measure your DNS latency and performance. Discussing the technical

details of these traffic monitoring tools is beyond the scope of this book, but if you want to learn more, see Microsoft's recommendations for [measuring traffic manager performance](#).

Azure CDN

Azure Content Delivery Network (CDN) is a global CDN solution for delivering high-bandwidth content by caching content in different locations. You can choose Azure CDN for applications hosted in Azure or any other servers or locations.

Azure CDN enables you to cache static objects coming from your web applications, Azure Blob storage, or a web server that is publicly available. It uses the closest POP server, as shown in [Figure 4-13](#), to accelerate dynamic content.

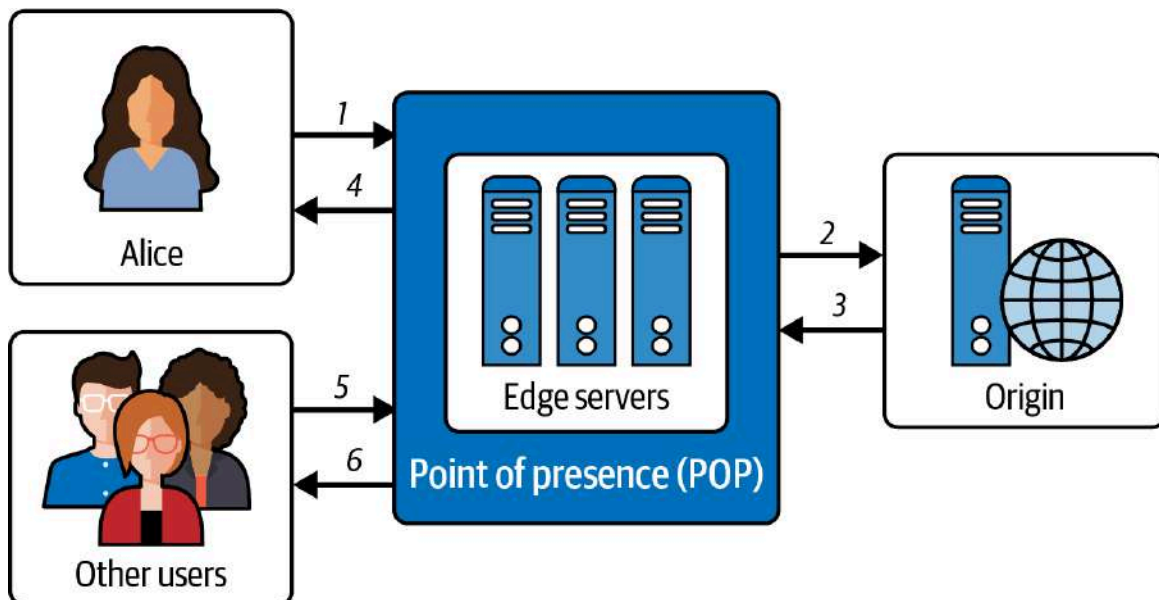


Figure 4-13. Point of presence server with Azure CDN

The following outlines the steps described in [Figure 4-13](#):

1. A user accesses an asset from a web application using Azure CDN, which routes the request to the best possible POP server location.
2. If there a cache of the requested asset on the POP edge servers, it will return it directly to the user.

3. If the asset requested is not cached, it will request it from an origin. An origin can be your Azure Storage or Azure Web App.
4. The edge servers will cache the file or asset and return the response to the user with requested asset.
5. Other users from different locations can access the same file or asset.
6. Since the file or asset is already cached on POP edge servers, it arrives faster back to these users.

Additionally, here are some of the features of Azure CDN:

- Better performance and user experience for end users globally
- Gain the ability to scale to handle instantaneous high load and demand
- Azure storage blobs can be used to cache content to make it publicly available
- CDN caching of web content, images, scripts, and other web assets
- Allows you to cache content based on specific query strings
- Access the cached content from a custom domain through CDN HTTP endpoint mapping

Azure CDN has different tiers such as Azure CDN Standard from Microsoft, Azure CDN Standard from Akamai, and Azure CDN Standard and Premium from Verizon. Prior to setting up your Azure CDN account, you should understand the different tiers and the **comparison of the different features they offer**.

Azure Networking Services for Network Monitoring

This section describes networking services in Azure that help monitor your network resources such as Azure Network Watcher and Azure Monitor Network Insights. We will take a look at each of these services and how to use them to monitor your network.

Azure Network Watcher

Azure Network Watcher is a composed of different tools used for monitoring, diagnosing, logging, and metrics management for an Azure Virtual Network. This networking service is commonly used to track and monitor network health statuses of IaaS services such as VMs, application gateways, load balancers, virtual networks, etc.

Following are some of the benefits of using Azure Network Watcher:

- Monitor communication between VMs and network endpoints at regular intervals with alerts and notifications
- Vizualize the overview and relationships of Azure resources in a virtual network
- Detect and troubleshoot any network traffic filtering problems between your VMs
- Diagnose problems of network routing and outbound connections for VMs
- VMs **packet capturing** that helps detect possible network anomalies

Azure Network Watcher is useful when you are troubleshooting issues related to detecting network traffic anomalies in Azure IaaS resources. For example, if you want to troubleshoot a problem with a VPN connection between two VMs, you can use Azure Network Watcher to monitor the traffic between them.

One limitation of using Azure Network Watcher is worth knowing. It will not work with web analytics and monitoring PaaS services because Network Watcher was not designed to be used with these two services.

Azure Monitor Network Insights

Azure Monitor Network Insights provides an overall view of health and metrics for all the network resources that you have deployed and hosted in

Azure. This Network Insights service is part of Azure Monitor and allows you to track networking metrics easily without advanced configuration.

It is structured around these key components of monitoring:

- *Network health and metrics* for the visualization of all your networking resources with the option to search, filter, and set up alerts and dependency views
- *Connectivity* helps you visualize all the configured tests done via **Connection Monitor**
- *Traffic* gives you a view of all flow logs for network security groups (NSGs) and also the traffic analytics for the selected set of subscriptions, grouped by location
- *Diagnostic Toolkit* can be used in troubleshooting network issues such as IP flow verification, packet capture, etc.

If you are using Azure Monitor for monitoring and gathering metrics for your Azure resources, then Network Insights is good tool to use, especially when you house multiple networking resources in Azure.

Azure Space: Networking Beyond the Clouds

Azure Space is one of Microsoft's most-advanced innovations. Microsoft wanted to create networking and connectivity beyond the clouds, to the space. It was created and serves as an ecosystem and **platform for the space community**.

As of this writing, there are three services created for Azure Space: Azure Orbital, Azure Modular Data Center, and Azure Orbital Emulator.

Azure Orbital

Azure Orbital is a fully managed ground station as a service (GSaaS). This service is currently in a preview version. It helps users to communicate, downlink, schedule service,

control their satellites, and scale operations from Azure. With the use of Azure Orbital, integrated data processing is easily managed on the Azure platform. Data being processed is transferred securely through the user's virtual network, which can be stored on Azure Storage or any Azure service. The main use of **Azure Orbital** is for global communications and earth observation. **Figure 4-14** illustrates how Azure Orbital is connected to Azure data centers using Azure WAN and how it connects to satellites in space using an orbital downlink.

Azure Orbital Emulator

Azure Orbital Emulator enables satellite developers to use AI algorithms to evaluate satellites before launching, and it can simulate complex satellite networking. The Orbital Emulator assists in getting perspective on how an application functions when it is in orbit. This tool enables satellite developers to test their applications in the cloud (Azure) first before they deploy them to space.

Azure Modular Datacenter

Azure Modular Datacenter (MDC) was designed and created to help users who need cloud computing solutions in hybrid, challenging environments and remote locations. As an MDC user, you have the capability to deploy your own self-contained transportable data center unit near the location you need it to be. In simple terms, Azure provides a portable data center unit (MDC unit) that is fully equipped with full network connectivity and secure network connectivity using satellite communications.

If your organization is interested in learning more about Azure Orbital and cloud services for space, check out their **Azure Space Partner Community** program.

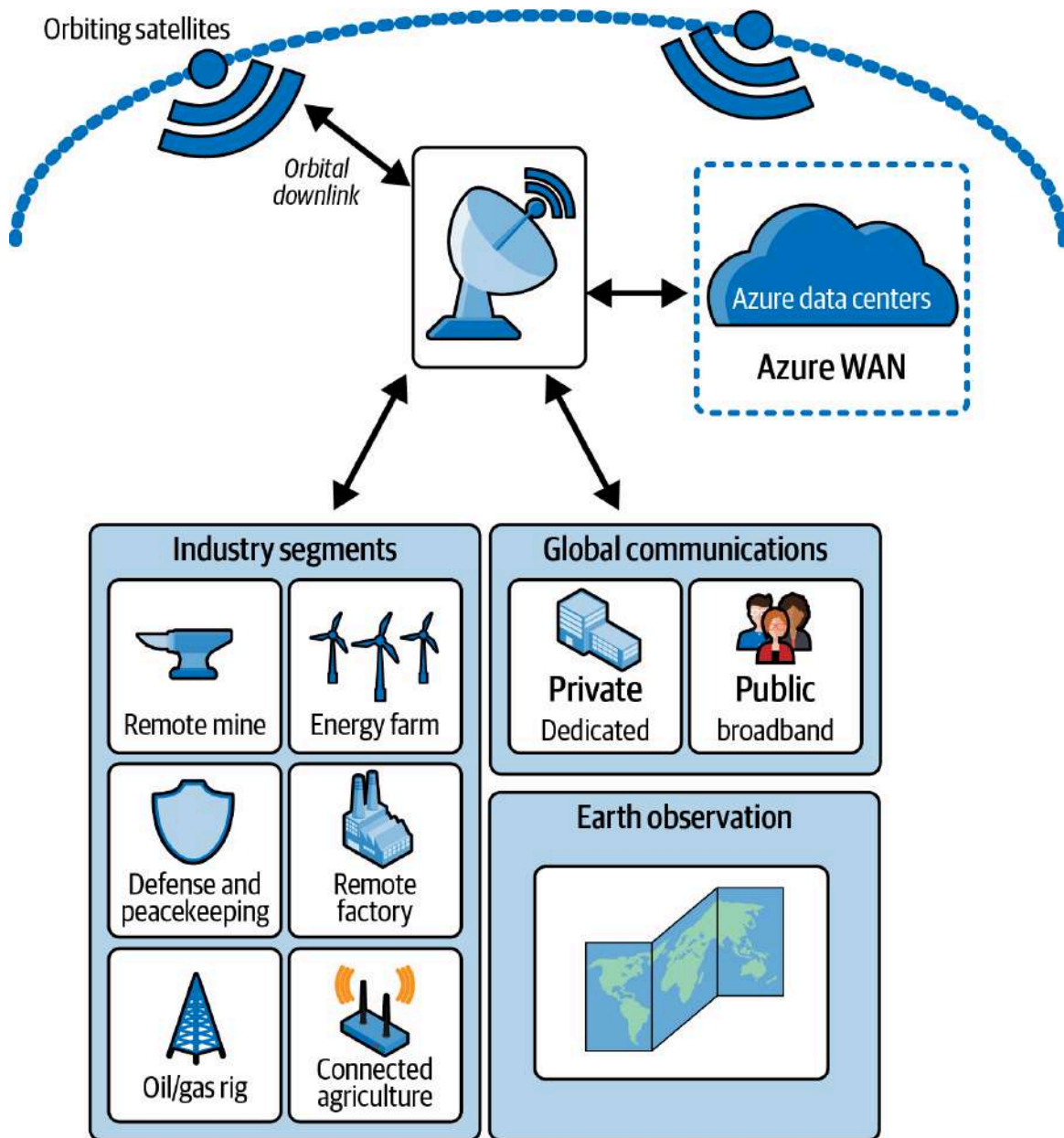


Figure 4-14. Global satellites and networks connect using Azure Orbital (adapted from an image by *Microsoft documentation*)

Learn By Doing (Try It!)

In addition to the hands-on lab [repository](#) for the topics in the this chapter, the following quick-start tutorials are recommended as they are updated based on Microsoft's technical updates for the service:

- [Quickstart: Create a virtual network using the Azure portal](#)

- Create a site-to-site connection in Azure Portal
- Quickstart: Create a public load balancer to load balance VMs using the Azure Portal
- Tutorial: Create a site-to-site VPN connection in the Azure Portal
- Quickstart: Create and modify an ExpressRoute circuit

Summary

In this chapter, we explored basic concepts, descriptions, features, and limitations of Azure's most common networking services and tools. We learned that networking services are categorized by their different purposes, such as connectivity, application delivery, security, and monitoring. Information on these networking services, grouped by category, provides an overview and ideas on which networking service to use along with compute resources in your cloud application development.

Azure has several load-balancing solutions, including Front Door, Traffic Manager, Application Gateway, which can be used individually or in combination. We also learned that Azure networking goes beyond the clouds of cloud computing to outer space through Azure Space services like Azure Orbital, Azure Orbital Emulator, and Azure Modular Datacenter that are offered as ground station as a service (GSaaS). GSaaS allows space explorers and researchers to communicate globally through space stations for **research and innovation**.

In the next chapter, you will learn more about the different storage and database solutions in Azure.

Check Your Knowledge

1. What is the basic block of a private network in Azure that enables your Azure resources to communicate with each other?

2. How can you protect and secure your applications in Azure using the networking resources available? (List a few examples.)
3. You need to securely connect your on-premises network to Azure in a private connection; which Azure networking services would you use?
4. What networking service would you use if you wanted to filter inbound and outbound traffic of Azure resources and secure your VNets?
5. What Azure networking service would you use if you want features like dynamic site acceleration, file compression, geo-filtering, and set caching rules for web content?

For the answers to these questions, see the [Appendix](#).

Recommended Learning Resources

“Architect Network Infrastructure in Azure.” Microsoft Learn, <https://oreil.ly/I2nQ->.

“Azure Front Door and CDN Documentation.” Microsoft Learn, <https://oreil.ly/lJGne>.

“Azure Networking Architecture Documentation.” Microsoft Learn, June 13, 2023, <https://oreil.ly/bnmIM>.

“Azure Networking Fundamentals Documentation.” Microsoft Learn, <https://oreil.ly/QOgMe>.

“Azure Orbital Documentation.” Microsoft Learn, <https://oreil.ly/payyz>.

“Enable Remote Work by Using Azure Networking Services.” Microsoft Learn, April 9, 2023, <https://oreil.ly/uNJzr>.

“Get Up and Running with Kubernetes.” Microsoft Azure, <https://oreil.ly/s3Vzj>.

Marczac, Adam. “AZ-900 Episode 10 | Networking Services | Virtual Network, VPN Gateway, CDN, Load Balancer, App GW.” Adam Marczak –

Azure for Everyone, YouTube video, August 18, 2020,
[*https://oreil.ly/eJmgP*](https://oreil.ly/eJmgP).

“Microsoft Azure Fundamentals: Describe Azure Architecture and Services.” Microsoft Learn, [*https://oreil.ly/XY5vc*](https://oreil.ly/XY5vc).

Savill, John. “Azure Master Class V2 – Module 6 – Networking.” John Savill’s Technical Training, YouTube video, January 17, 2023,
[*https://oreil.ly/q4-Fx*](https://oreil.ly/q4-Fx).

Smith, Derek. “Zero to Hero with Azure Virtual WAN by Derek Smith.” MC2MC, YouTube Video, December 20, 2021, [*https://oreil.ly/Xn9Gv*](https://oreil.ly/Xn9Gv).

Smith, Derek. “Azure Network Engineer Associate (AZ-700) Cert Prep: Secure Network Connectivity to Azure Resources.” LinkedIn Learning, September 25, 2023, [*https://oreil.ly/DC82Y*](https://oreil.ly/DC82Y).

Valiramani, Avinash. Microsoft Azure Networking: The Definitive Guide. Microsoft Press, 2022.

“Why Use Azure Orbital Ground Station?” Microsoft Learn, August 15, 2023, [*https://oreil.ly/qYhet*](https://oreil.ly/qYhet).

Chapter 5. Microsoft Azure Cloud Storage and Databases

At the heart of any application is the data, and whether that data is an image, an arbitrary fact about lumpia, or something in between, it needs to be stored somewhere! Luckily, there are plenty of data storage and database options that suit specific needs. Which do you use? That will depend on some key factors, like how you intend to interact with your data, how much of it you need to store, what kind of data you are storing, and how your application is structured. In the end, choosing the right database and data storage options can mean the difference between a resilient, scalable, and delightful application...and one that is not, so choose wisely!

—Adrienne Braganza Tacke, Senior Developer Advocate
and Software Engineering Leader at Cisco

In the previous chapter, you learned about the importance of networking in the cloud infrastructure of Microsoft and the various Azure networking services for different purposes. Implementing database services like relational databases and Microsoft SQL Services has been a key part of software development with the Microsoft technology stack. ¹

In this chapter, you will learn about the benefits of cloud storage in Azure and the different cloud database services for different types of data. Cloud storage services are important in daily practical data management, software development, and cloud engineering. By the end of this chapter, you will understand the different data storage and database solutions you can use for your web applications and data storage hosted in Microsoft Azure.

Data Storage and Databases in the Cloud

Having effective strategies for storing data for enterprise applications and systems is critical and vital for the success of businesses and organizations.

Cloud computing has been helpful to many organizations moving to digitalization, and cloud solutions have helped bioscience research, innovation, and collaboration.² Cloud computing aids in collaborative research by providing secure access to important digital tools and data management systems with powerful, reliable, and scalable computing resources.

The volume of enterprise data is growing exponentially; the data we store, e.g., analytical data, financial data, client data, vendor data, etc., continues to expand when our businesses grow. Therefore, businesses need a sustainable data storage solution that can handle the data they need to store.

Storing data in the cloud provides the benefits of highly scalable, secure storage for business applications, systems, and workloads that will also increase cost-effectiveness in the long run.

Database administrators, data engineers, data scientists, and data analysts play vital roles in data-driven organizations, society, and cloud solutions. These professionals engineer and manage the enterprise data stored in digital storage or databases through systems and applications.

If you are a data professional working with data in your job, reading this chapter will help you accelerate your skills in data engineering and cloud development by learning more about Azure's solutions for data storage and databases.

Data Storage Management in the Cloud

Users of your systems or applications must be able to access and manage their data whenever and wherever they are. Therefore, storing data in any cloud platform, like Azure, has been an effective solution and option for data storage management. Another important aspect of data storage management is the policy for data retention.

A data retention policy helps organizations define which data they need to retain for security, compliance, and operational reasons. These retention policies are important to have, especially when business data continues to grow exponentially. Storage management for data, either structured or unstructured, will assist in resource provisioning, configuration, cost management, and evaluation.

Using Microsoft Azure as a cloud storage provider, you can easily manage your data storage and retention policies for your Azure resources. For example, if you are using Azure Storage, you can set up **time-based retention policies for immutable blob data**.

Benefits of Digital Storage in the Cloud

There are many reasons cloud storage is beneficial to businesses.³ Storing data digitally and in the cloud is efficient and convenient. Cloud storage allows users to minimize using a physical local hard disk or storage that is at risk of getting lost. Storing data digitally enables global and remote access with dynamic options to upgrade storage capacity on demand when we need it.

Enterprises and businesses can save money⁴ by using cloud storage services. They incur expenses for storing their data in their own servers or data centers. Cloud computing providers offer storage services with security, reliability, and scalability features at a global scale.

Another reason cloud storage is beneficial is the synchronization of data securely stored on the cloud, which makes it easier for users to share and access important files or content across any device from anywhere. Aside from the flexibility of upgrading your storage anytime, there is also an advantage of better storage management with better data backup management and disaster recovery benefits that the cloud provider offers. So in case of an emergency data loss, you can retrieve and restore your backup data from the cloud.

Big Data, Structured Databases, and Non-Structured Databases

The term *big data* is commonly used to describe large volumes of data that can be helpful for businesses but difficult to manage. For example, data can be used to analyze and improve the customer service strategy of a business.⁵

In addition, these days, the data that we generate and that is collected through our applications in the cloud is enormous and growing, which means we need to find ways to handle the complexity and heterogeneity of this data. Using traditional relational databases create challenges in handling such massive amounts of unstructured.

To understand the importance of big data, we need to understand the different types of data that we can store in the cloud. The different data types are structured and unstructured data.

Structured data

Suppose you are a developer working with a Microsoft programming language like .NET. In that case, you have developed or have been developing applications with SQL databases for gathering data for different types of CRUD operations. Many past and present systems and applications have managed databases using SQL, which is structured, organized, and formatted to be easily searchable in relational databases. These databases are in the form of relational databases with tables that relate to each other. Constraints and relationships between the data in these tables can be one-to-many or many-to-many. SQL Server, Oracle, MySQL, Azure SQL, Postgres, and many more are relational database systems commonly used today.

Non-structured data

Non-structured data does not have a predefined format. Usually, this type of data is qualitative data like video files, text files, audio files, mobile activity history logs, posts on social media, images from a satellite, etc. Non-structured data cannot be organized and structured as on typical relational databases like SQL. Databases suitable for storing and managing non-structured data are non-relational or NoSQL databases. Certain projects, especially open source projects, use databases with NoSQL data structures, for example databases for key-value pairs, documents, graphs, and other non-structured data collection. **Azure Data Lake** can also be used to analyze and manage unstructured data.

Understanding the different data types allows you to easily identify and decide on the right Azure cloud storage or databases, regardless if they are stored on existing on-premises databases or already hosted on the cloud.

Figure 5-1 illustrates the difference between structured and non-structured databases.

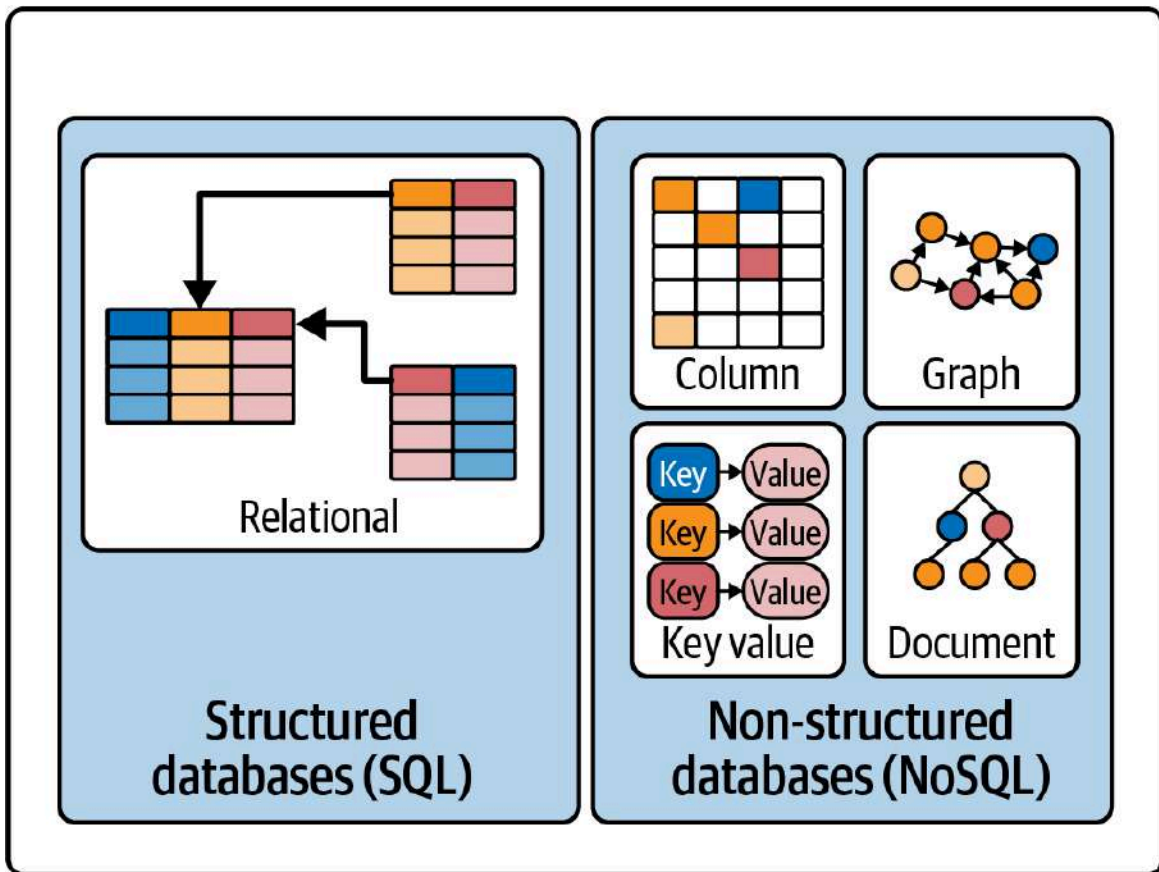


Figure 5-1. Structured versus non-structured databases

NOTE

Structured data is quantitative data that is highly organized and easily searchable in traditional relational databases. Unstructured data is qualitative and has no specific predefined format or structure. Most of the time, structured data is easier to collect, process, and analyze compared to unstructured data.

Understanding whether the data is structured or non-structured is important, not just from a business and knowledge perspective but also to determine the ideal type of cloud storage service or database solution you need when storing and managing this data on a cloud platform such as Microsoft Azure.

Azure Storage and Database Services in the Cloud

Microsoft has a wide variety of **cloud storage solutions** for different types of data storage scenarios. Applications and systems need storage and databases to store, manage, analyze, and visualize enterprise and application data. Azure solutions are available for storage, databases, visualization, and data analytics.

Azure Storage

Azure Storage is a data storage solution that allows users to store and manage their data in the cloud. Moreover, it provides durable, secure, scalable, and highly available storage via HTTP, HTTPS, or REST API for any type of data of any size in your organization. The Azure Storage service has different account types: Azure Storage v1 and v2, which have **distinct capabilities and features** that are worth checking out, especially if you're in the process of choosing which type of cloud storage you need for your use case.

Furthermore, the Azure storage platform ensures that the data stored in the cloud is available in case of unexpected system failure or emergency. Users can replicate their data across different geographic locations and data centers for backup and disaster recovery.

Azure Storage is built to be scalable, able to handle the performance, capacity, and storage demands of your organization. Using Azure Portal and the free cloud storage tool **Azure Storage Explorer** you can easily administrate your data in Azure Storage from the cloud directly from your computer on any operating system.

IT professionals, data engineers, and developers can access productive tools to use with Azure Storage, such as client libraries to build APIs and web services with the Azure Storage platform using the various supported languages like C#, Java, Python, C++, and TypeScript. Cloud engineers and

Azure developers can also build scripts, configuration, and automation tasks using tools in Azure like [Azure PowerShell](#) and [Azure CLI](#).

The different types of Azure Storage data services and their specific use cases will be discussed in detail in the later sections of this chapter.

Database Services in Azure

Aside from Azure Storage there are other fully managed solutions to handle the data we collect and store from and to our applications. Azure provides different types of cloud database services that are powerful, reliable, and scalable. There are different options for both relational, non-relational (NoSQL), and in-memory database solutions available for modern cloud application development use cases for your organization.

Services for Azure Storage

In this section of the book, you will learn about the different types of services available for cloud storage depending on the type of data your organization needs to store and manage. [Table 5-1](#) shows some of the common Azure Storage services for different use cases and the type and size of data you are trying to collect and store. Please note that these Azure Storage services may change overtime. To keep track of each of its respective feature updates, visit the [Azure updates website](#).

Table 5-1. Azure Storage data service options

Name	Description and Purpose
Azure Blob Storage	If you need a massive object store that should be scalable then storing blobs (for example, text and binary data) with Azure Blobs is ideal. It is ideal for example if you want to store and manage your blob files on the cloud on any accessible web browsers on any platform for distributed access for external users or applications. Big data analytics through Data Lake Storage as well as backup and recovery solutions are also supported in this service.
Azure Files	This service is useful if you need a file storage service for storing and managing files on any accessible web browsers on any platform for distributed access for external users or applications. Azure Files is a file share in Azure that is fully managed and can be accessed using standard protocols like Network File System (NFS) and Server Message Block (SMB) .
Azure Queues	If you need to securely store large numbers of messages between applications, Azure Queues enables you to store messages in queues to be processed between components of your applications. These queue messages can be accessed securely using the REST, HTTP, or HTTPS protocols.
Azure Tables	If you need a non-structured or schemaless store of your structured data that can be accessed internally and externally via the cloud infrastructure in Azure then Azure Tables is ideal.

Name	Description and Purpose
Azure Managed Disks	If you want to handle and manage large block-level storage volumes or if your workloads need support for virtual disk features and storage for infrastructure as a service (IaaS) or virtual machine deployment.

Azure Blob Storage

Azure Blob Storage is a cloud-based object storage service that allows you to store and manage your Binary Large Object (blob) data in the cloud. This service is able to handle massive text and binary data, especially unstructured types.

Some of the common uses of Azure Blob Storage are managing images, videos, or files on any web browser and device, and if you need to store files for external distribution or access either for business purposes or personal use.

If you want to stream blob files like videos or audio files for your video streaming services or podcast, you can also use the Azure Blob Storage service to access this data from your application or directly from your browser. Another important use is performing backup and archiving of data.

One of the most important concepts to know when you are setting up your Azure Blob Storage is its different components. Blob Storage has three types of resources: the Azure storage account, the container in this storage account, and then the data object or blob items that you store in this container.

Components of Azure Blob Storage include:

Storage account

You need a storage account because it is a unique namespace for your blob data. An Azure Storage account

provides a unique URL address or link to every data object in your storage. Usually when you set up or create your storage account, you need to decide and choose a unique account name. The Azure Storage blob endpoint and account storage name becomes the base link address for the data objects in your storage account. If your storage account is named *learningmicrosoftazurestorage*, then the default endpoint link for blob storage is *learningmicrosoftazurestorage.blob.core.windows.net*.

Containers

In an Azure Storage account, you can have an unlimited number of containers in a storage account. In each container, there is no limitation to the number of data objects or blobs you can store. Containers are where you organize your set of blobs just like you organize photos or videos in a standard file system. Basically, containers are like traditional digital folders on computers, resembling a file system directory.

Blobs

Azure Storage supports *block blobs*, *append blobs* and *page blobs*. Block blobs are used to store text and binary data that can be individually handled. They can store blobs up to 190.7 TiB. Append blobs are composed of block blobs specially used for append operations. These blobs can be used for scenarios like VM data logging. Page blobs are useful in storing random access files up to 8 TB; for example, it can store a virtual hard drive (VHD) for Azure VMs.

The data objects you store in Azure Blob Storage are accessible anywhere worldwide through secured and authenticated protocols like HTTPS. Your development team or organization can implement application solutions with Azure Blob Storage using the development tools and technologies

supported, such as client libraries for different programming languages like .NET, Java, Python, Node.js, Ruby, PHP, etc. Data can be accessed securely using REST APIs, Azure CLI, and Azure PowerShell.

WARNING

To find out more about naming conventions and best practices when working with Azure Blob Storage, check out [Microsoft's documentation](#).

Azure Files

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard **Server Message Block (SMB) protocol** or **Network File System (NFS) protocol**.

Azure Files file shares can be mounted concurrently by cloud or on-premises deployments. SMB Azure file shares are accessible from Windows, Linux, and macOS clients. The NFS Azure file shares are accessible from Linux or macOS clients. Additionally, SMB Azure file shares can be cached on Windows Servers with Azure File Sync for fast access near where the data is being used.

Table 5-2 shows some of the common use cases and benefits of using Azure Files.

Table 5-2. Use cases and benefits of Azure Files

Uses and Benefits	Description
Replacement of on-premises file servers	For easy sharing and global access Azure Files can be used as an alternative to network-attached storage (NAS) or traditional storage on premises.
Flexible option to migrate applications	It is easy to “lift and shift” applications to the cloud that expect a file share to store file application or user data. Azure Files supports both the classic lift and shift and hybrid lift and shift scenarios.
Simplify cloud development	Run stateful and long-running background jobs or tasks with autoscaling capabilities based on CPU or memory usage.
Persistent volume for containerization	Containers deliver build once, run anywhere capabilities that enable developers to accelerate innovation. For the containers that access raw data at every start, a shared file system is required to allow these containers to access the file system no matter which instance they run on.

Azure Queue Storage

Azure Queue Storage is a storage service for storing large numbers of messages in the cloud that you can access and manage anywhere securely. A queue message can be up to 64 KB and can contain millions of messages depending on the storage capacity allowed on your account.

Following are key components in Azure Queue Storage:

URL format

The URL format of Azure Queue Storage is endpoint or URL in this format: *storageaccountname.queue.core.windows.net*. If you want to access an image name called *my-image* from a queue storage, you can access it using the following address: *storageaccountname.queue.core.windows.net/my-image*.

Storage account

An **Azure Storage account** stores the data objects for your queue messages. The storage account is a unique namespace that allows you to access your storage anywhere via secured protocols over the internet like HTTP or HTTPS. An Azure Storage account is known for its durability, security, scalability, and high availability.

Queue

A set of message can be contained in a queue. When setting up names for your queue storage there are **naming conventions** to be mindful of—for example, the queue name must be in all lowercase.

Message

A message in Azure Queue Storage can be in any format (up to 64 KB). Before July 29, 2017, the allowed maximum **time-to-live (TTL)** was seven days; for later versions the maximum TTL can be any positive number, or -1 indicating that the message doesn't expire. If you do not set this TTL parameter, the default is seven days.

Azure Table Storage

If your organization needs to collect, store, and manage large amounts of structured data, then Azure Table Storage is ideal for handling it. Azure Table Storage is a NoSQL data storage capable of internal and external

authenticated calls from and to Azure. It also is ideal for storing structured data or relational data.

You can use Azure Table Storage for:

- Collection and storage of terabytes (TBs) of structured data for web applications
- Data access using the **OData protocol** and **LINQ queries** for **WCF Data Services in .NET Libraries**
- Storage that is easy to denormalize and non-complex datasets
- Quick data queries using clustered indexing

Azure Table Storage includes the following components:

Storage account

All access to Azure Storage is done through a storage account. All access to Azure Cosmos DB is done through a Table API account.

URL format

Azure Table Storage accounts use this URL format: *mystorageaccount.table.core.windows.net*. If you are using the Azure Cosmos DB Table API account with this service, the URL format is *mystorageaccount.table.cosmosdb.azure.com*. Using the OData protocol you can also directly access Azure Table Storage using these URL formats.

Table

A table is a collection of entities that doesn't implement and enforce a schema. A table can have entities with different sets of properties.

Properties

A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has three system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.

Azure Table Storage will automatically scale as demand increases especially if you use it for storing and querying large sets non-relational or structured data.

WORK WITH THE DATA IN AZURE STORAGE USING AZURE STORAGE EXPLORER

To manage your data or files easily on your Azure Storage account, use the free stand-alone app [Azure Storage Explorer](#) on Windows, macOS, and Linux platforms.

Azure Managed Disks

Azure Managed Disks are a recommended solution if you have block-level storage volumes for IaaS Azure VMs and Azure VMware solutions. This cloud storage solution is made up of virtualized disks on the cloud. Think of it as a virtualized version of your on-premises physical storage. These virtual managed disks in Azure are fully managed, which means that when you set them up, you need to choose some of the properties, such as the size, the location, the type of the disk, e.g., ultra disks, standard solid-state drives (SSDs), premium SSDs, etc., and the operating system you want to install on the disk.

Table 5-3 describes the uses and benefits of Azure Managed Disks.

Table 5-3. Uses and benefits of Azure Managed Disks

Uses and benefits	Description
Availability, high durability and scalability	Managed disks assure and secure several data replications, giving high durability and availability of 99.999% in case of data loss or failures.
Availability sets and availability zones integration	Availability sets are supported and can be integrated with Azure Managed Disks to solve the problem of single point of failure through isolation. Azure availability zones also protect your resources from failures in the data center.
Easy and flexible VM deployments	Managed disks enable you to create up to 50,000 VM disks per Azure subscription per region. Azure VM Scale Sets scale your VM workloads while allowing you to deploy thousands of virtual machines in a scale set alone, which helps you save money.
Secured	Managed disks are designed with security by default using RBAC (role-based access control) to protect your data from unauthorized access, and the storage is encrypted e.g., server-side encryption (SSE) and Azure Disk Encryption (ADE)

Azure Storage Security Best Practice Tips

Azure Storage in cloud development with any platform or programming language uses **shared access signatures (SAS)** to access the storage account

from your apps or code. However, using SAS for Azure Storage access can create major security risks.

For example, if an SAS token is compromised, the attacker can access your data. Therefore, it is important to be aware of the best security practices when using SAS for Azure Storage access.

Top security recommendations for Azure Storage include always using HTTPS to distribute SAS, setting up an SAS token expiration policy, and always implementing **user delegation for SAS** whenever possible.

Check out a list of **recommended storage security best practices** and the **best practices on using SAS with Azure Storage** in Microsoft's documentation.

Azure Database Services

In the earlier part of this chapter, you learned the common uses and benefits of cloud storage for users and organizations. However, many businesses and organizations still have challenges with data platform strategies for their systems or applications. For example, some organizations have outdated data platforms that need upgrades to most effectively integrate with cloud services.

The current trend is to move existing systems to the cloud, build new applications quickly with the cloud, and offload some on-premises costs. However, to do this effectively, you need a plan for how to move workloads to the cloud. You also need to understand how the role of a database administrator (DBA) or data professional stays the same and what changes you'll have to make.

Azure SQL as a Fully Managed Database Service

Azure SQL Database cloud database service is always up to date and is a fully managed relational database service developed and built for cloud computing. Using Azure SQL Database, you are able to create SQL databases with key features and capabilities for your applications.

Azure SQL deployment options

There are different deployment options available for Azure SQL. Depending on your current workloads and infrastructure, you can choose to deploy your relational databases in different ways:

SQL Server on Azure Virtual Machines

SQL Server on a virtual machine is a version of SQL Server that runs in an Azure VM and is categorized as an IaaS. It's an SQL Server instance, so all your SQL Server skills should directly transfer, though Azure can help automate backups and security patches. You're responsible for updating and patching the OS and SQL Server, apart from critical SQL Server security patches.

SQL Managed Instance

SQL Managed Instance is a PaaS deployment option of Azure SQL that allows organizations to save money on virtual machines or servers. It helps by giving users an instance of SQL Server but removes much of the overhead of managing a virtual machine. Most of the features available in SQL Server are available in SQL Managed Instance. This option is ideal for customers who want to use instance-scoped features and want to move to Azure without rearchitecting their applications.

SQL Database

SQL Database is a PaaS deployment option of Azure SQL that abstracts both the OS and the SQL Server instance away from users. This deployment option allows you to get a database and start developing applications. SQL Database is the only deployment option that supports scenarios that require unlimited database storage (hyperscale), supports geo-active replication, automatic failover groups, and autoscaling for unpredictable workloads (serverless). SQL

Database has the industry's highest availability SLA. It provides other intelligent capabilities related to monitoring and performance, partly because Microsoft manages instances.

NOTE

In 2008, when Microsoft Azure (previously called Windows Azure) launched, one of its top key components was Microsoft SQL Services. This was eventually renamed to Azure SQL. SQL Server and Azure SQL continually evolved with better features that expanded to support and other open source databases like PostgreSQL, MariaDB, and MongoDB.

Enterprise systems are rapidly evolving, which means users need to handle and manage large volumes of assorted and diverse data in various formats and types. Therefore, choosing the right data store for your business data is a critical decision.

Check out Microsoft's [guide to choosing the right data store](#) and the [data technology choices](#) offered and discussed as part of the *Microsoft Cloud Adoption Framework for Azure*.

Azure Cosmos DB

Azure Cosmos DB is a fully managed NoSQL database for modern app development. It takes database administration off your hands with automatic management, updates, and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

Solutions that benefit from Azure Cosmos DB include web, mobile, gaming, and IoT applications that need to handle massive amounts of data, reads, and writes at a global scale with near-real response times for a variety of data. These applications will benefit from Cosmos DB's guaranteed high availability, high throughput, low latency, and tunable consistency, as shown in [Figure 5-2](#).

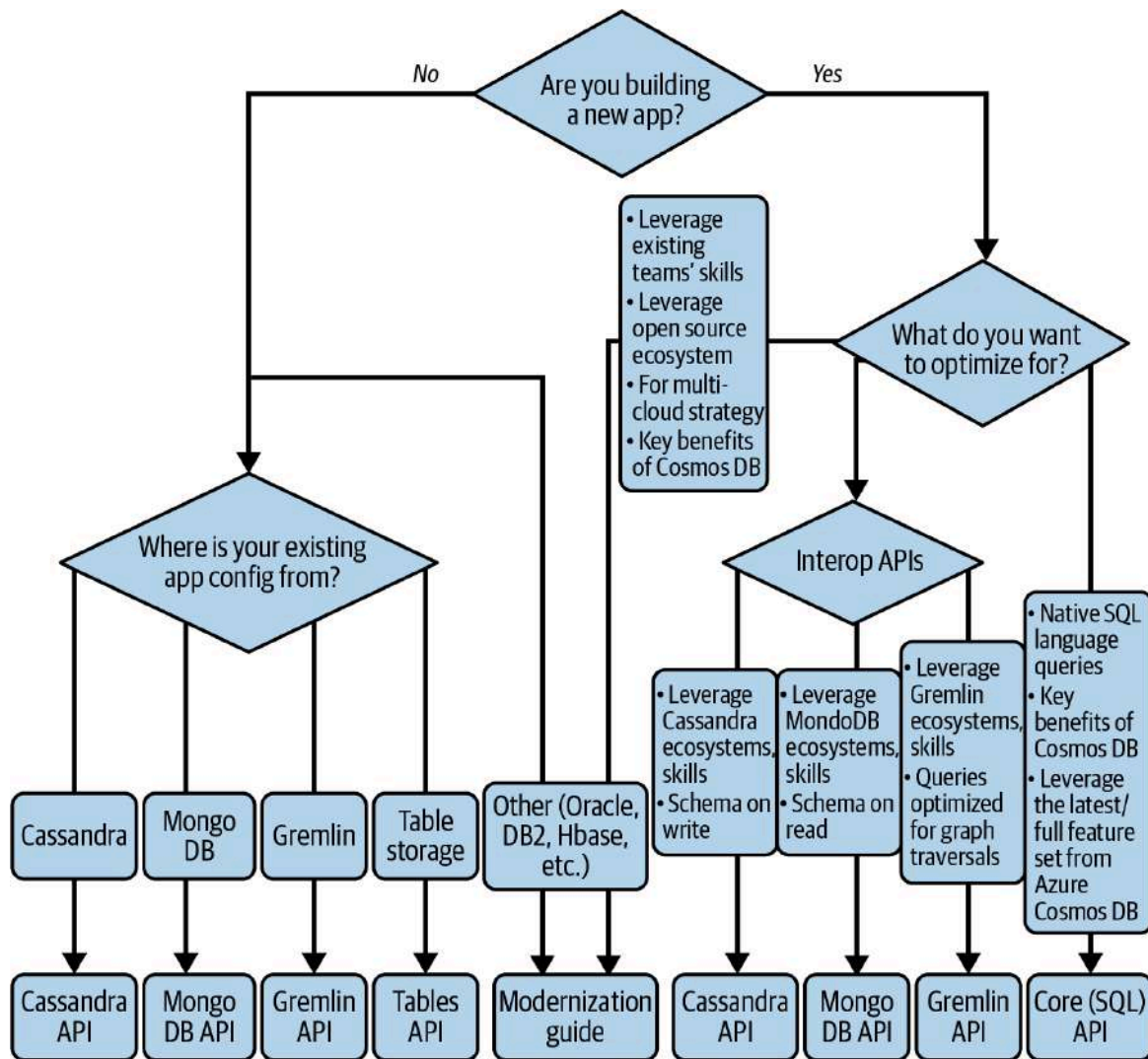


Figure 5-2. Recommended decision tree for choosing Azure Cosmos DB solutions

Following are benefits of using Azure Cosmos DB for your data solutions:

Guaranteed scalable speed

Gain unparalleled **SLA-backed speed and throughput** for global access, and instant speed and elasticity with SLA support. There is also support for multi-region writes and data distribution to any Azure region.

Easier app development

Development is easier and more efficient with open source APIs and SDKs supported by Azure Cosmos DB. You have the option to integrate with other Azure services commonly used in cloud-native app development like Azure Functions, IoT services, Azure Kubernetes Service, App Service, etc. Different programming languages are also supported, for example, SDKs for .NET, Node.js, Java, and Python.

Mission-critical ready

Azure Cosmos DB promises high availability and enterprise-level security. Azure provides the option to easily distribute and replicate your data to any Azure region.

Fully managed and cost-effective

Using the Azure Cosmos DB service on Azure is cost-effective as it is a fully managed cloud service. It is easy to work with database management that you can do on demand using the Azure Portal, Azure CLI, etc. Azure Cosmos DB also supports the Serverless (Consumption Plan) pricing model.

Azure Cosmos DB provides many ways to store data depending on what organizations need to build, create, and deliver.

Azure Cosmos DB consistency levels

A common confusion for those just getting started in Azure Cosmos DB is understanding its different consistency levels. To make sense of it you need to be familiar with the concepts of **consistency levels**, **replicas**, and throughput in Azure Cosmos DB.

For **distributed databases**, consistency is an important concept to understand especially when working with managed cloud databases like Azure Cosmos DB. Distributed cloud databases are operational across different physical locations. Furthermore, distributed databases depend on replication

configuration, usually for the purposes of low latency and high availability or both.

Depending on the priorities and requirements, there is a trade-off between choosing read consistency, availability, latency, and throughput in distributed cloud databases, as defined by the **PACELC theorem**.

For distributed cloud databases, *replicas* are copies of source data. These replicas can be hosted in different Azure regions. Usually the replica is configured for disaster recovery and geo-based distribution purposes.

If you are replicating your distributed databases in the cloud in Azure regions located away from your users, there might be issues with performance, response time, and latency. Therefore, when configuring the replication, it is ideal to choose the right consistency level for your needs.

Technical details of the different consistency levels (strong, bounded staleness, session, consistent prefix, and eventual) are beyond the scope of this book; however, if you want a deep dive into what is best for your distributed database workloads, refer to **Microsoft's documentation about consistency levels**.

NOTE

Azure Synapse Link for Azure Cosmos DB is an optional integration that can be used to take advantage of cloud-native hybrid transactional and analytical processing (HTAP) capabilities that enable real-time analytics over operational data in Azure Cosmos DB.

You can gain cost savings on Azure Cosmos DB using data retention with time to live (TTL) and the free tier. When you use Azure Cosmos DB you do not have to reserve your storage in advance because you get billed for the storage you have consumed directly. However, use cases that require a lot of writes might have an effect on the costs. To save costs, you can set a TTL for your data. This is a time interval after which the data will be deleted. This means the data you have stored elsewhere can be immediately

purged. This can help keep your relevant data updated. Learn more about [TTL on Azure Cosmos DB in the documentation](#).

You can also use the [Azure Cosmos DB free tier](#) if you are just getting started with the service, or trying to develop small production application workloads for testing.

Azure Cosmos DB APIs

Azure Cosmos DB has several options for database APIs, including the Core (SQL) API, Gremlin API, MongoDB API, Cassandra API, and Table API.

These Cosmos DB database APIs helps store and manage real-world data such as tables, key-value data, graphs, documents, and column-family data models.

The following database APIs enable database engineers, developers, and organizations to save effort and resources in adopting Azure Cosmos DB. Using these APIs your existing tools, systems, and skillsets can easily transfer and you can use the same data modeling and querying techniques in databases where the existing data is stored.

Azure Cosmos DB APIs are fully managed service in Azure's cloud platform. These APIs have features like throughput and storage automatic scaling, guaranteeing flexibility and performance because of their global scaling features. APIs currently available for Azure Cosmos DB include:

Core (SQL) API

Using the Core (SQL) API, you can store data in a document format and fully manage it with an end-to-end experience and SDK client libraries. The Azure Cosmos DB for SQL API accounts allow database developers and engineers to query database items using the SQL syntax. This API is ideal if you are migrating from databases such as Oracle, HBase, DynamoDB, etc. Also, if you want to utilize modern technologies to build your apps, SQL API is the recommended option. SQL API supports analytics and offers

performance isolation between operational and analytical workloads.

Gremlin API

The Gremlin API is worth considering for graph queries and if you need to store data as edges and vertices. This API is recommended for scenarios involving dynamic data with relations that are too complex to be modeled with the standard relational SQL databases. The Gremlin API for Azure Cosmos DB combines the features of the algorithms of graph databases with highly scalable, managed infrastructure. This API is based on the **Apache TinkerPop** graph computing framework and uses the partition strategy of Azure Cosmos DB for read and write operations. Use cases for the Gremlin API include geospatial data, social network graphs, data from Internet of Things (IoT) devices, and any devices that need to be represented graphically.

API for MongoDB

The API for MongoDB is specific to **MongoDB** databases, which store data in a document structure, typically in **BSON format**. Using the API for MongoDB is ideal if you want to work within the MongoDB ecosystem. If you already have an existing app with MongoDB, you just update the connection configurations and migrate existing data using native MongoDB tools or the **Azure Database Migration Service**.

Cassandra API

The Cassandra API is useful if you want to store data written for **Apache Cassandra**, which is usually data in column-oriented schema. It enables horizontal scaling in storing large volumes of data and offers a flexible approach to a column-oriented database schema. Use this API if your Cassandra databases need elasticity and you want to take

advantage of the fully managed features of Azure Cosmos DB. You can use the native Apache Cassandra features, tools, and ecosystem when you use this API, which means you don't need to manage infrastructure like Java VM, garbage collection, etc. Apache Cassandra client drivers are also supported to connect to the Cassandra API.

Table API

The Table API in Azure Cosmos DB stores data in key-value pair format and supports only **OLTP** scenarios. Any application written for Azure Table storage can be migrated to the Table API with minimal code changes and can also use the premium capabilities.

Azure Cosmos DB and its available APIs provide flexibility in choosing the appropriate database API for your needs. The next section discusses another interesting feature that Azure Cosmos DB offers for distributed database workloads.

Global distribution and replication using Azure Cosmos DB

Azure Cosmos DB is a fully managed and globally distributed database system that allows you to read and write data from the local replication of your databases. It enables replication of data to all the regions associated with your Cosmos account.

This cloud database service in Azure delivers elastic scalability of throughput, low latency, well-defined semantics for data consistency and high availability. Therefore, if your application needs fast response time in any location or region in the world, and if it's required to be always online, then building modern applications with Azure Cosmos DB is ideal.

A set of databases can be configured to be globally available and distributed in any of the Azure regions. To strategically lower latency, you should provision your databases to the Azure region closest to your application or

database user. Azure Cosmos DB replicates the data to all the Azure regions that are associated and linked with your Cosmos DB account.

Azure Cosmos DB also gives you the option to add or remove the regions associated with your account when needed. When you update the region for your account, your application doesn't need to be paused or redeployed. Cosmos DB is available in all five distinct **Azure cloud environments** available to customers.

Figure 5-3 shows how the distribution of Azure Cosmos DB is replicated and set up globally.

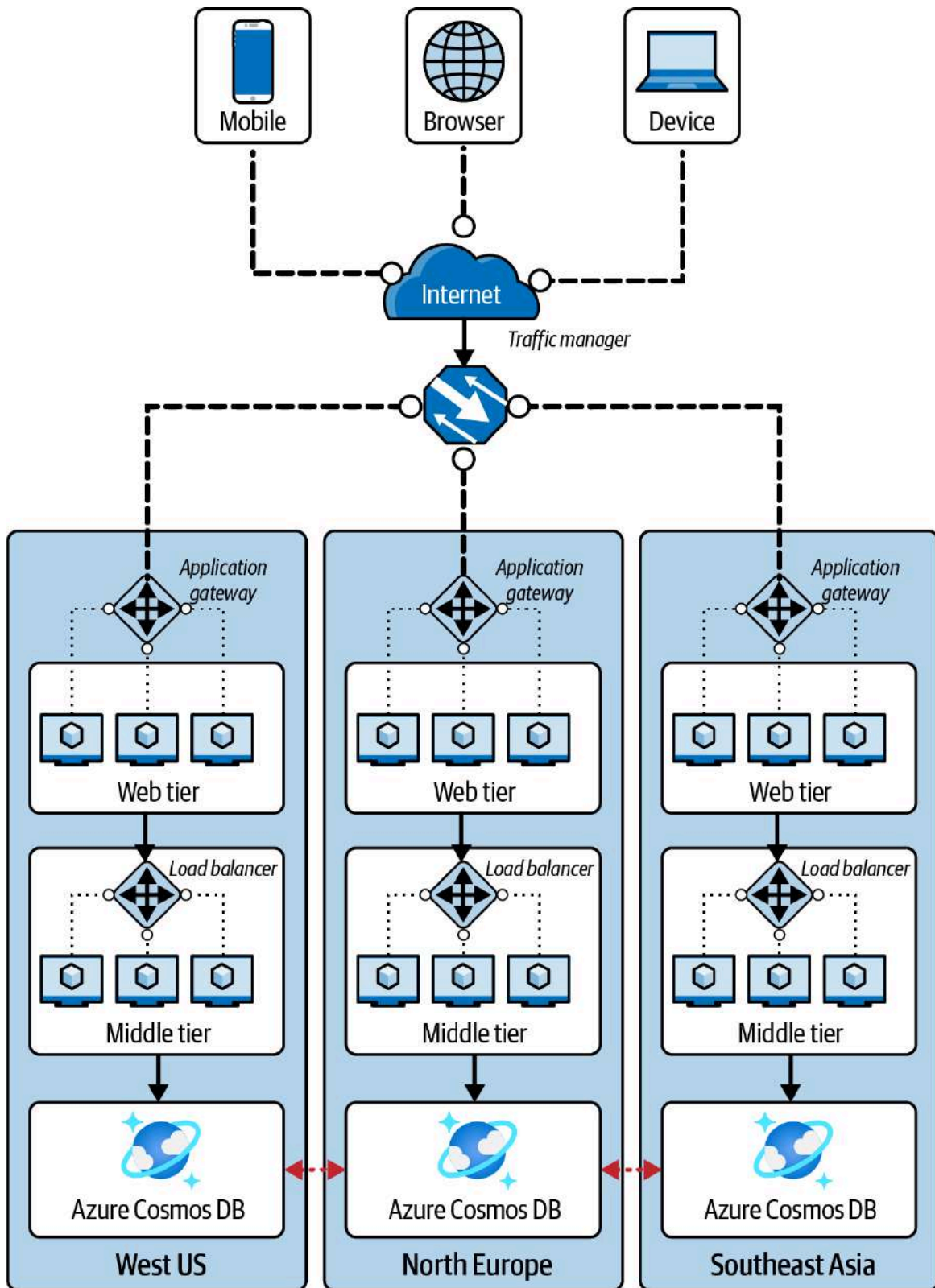


Figure 5-3. Global distribution using Azure Cosmos DB

TIP

If you need to migrate or import data from external sources into a container in Azure Cosmos DB, check out the Azure Cosmos DB Data Migration tool, which allows you to import data from sources in file formats, including CSV, JSON, XML, SQL, Azure Table Storage, MongoDB, Amazon's DynamoDB, and even other Azure Cosmos DB API collections.

For detailed information, check out [Microsoft's documentation on migrating data](#).

Deploying Azure Cosmos DB as infrastructure as code (IaC) is also possible. Azure enables developers and cloud engineers to automate deployments integrating Azure Bicep to manage Azure Cosmos DB resources. To learn more see Microsoft's guide on managing [Azure Cosmos DB Core \(SQL\) API resources with Bicep](#).

Azure Cosmos DB has features for any type of data. Choosing the right data store or database in Azure is an important step regardless if you plan to upgrade your existing enterprise systems or build a new application hosted on the cloud.

NOTE

Azure Cosmos DB gives us different API options, which helps us easily migrate new or existing data to Azure. However, every business requirement is unique so check out the ["Choose an API in Azure Cosmos DB"](#) guide if you and your team are uncertain about which API to choose.

Learn By Doing (Try It!)

Aside from the supplementary learning hands-on lab repository for the topics in this chapter, the following tutorials are recommended as they are updated based on Microsoft's technical updates for the service:

- [How to create a storage account](#)

- Quickstart: Upload, download, and list blobs with the Azure portal
- Quickstart: Azure Cosmos DB for NoSQL client library for .NET
- Quickstart: Azure Cosmos DB for Apache Gremlin library for .NET
- Microsoft Azure Fundamentals: Describe Azure architecture and services

Summary

In this chapter, we learned about the different technologies in Azure for storage and database solutions in the cloud. We learned about Azure Storage, which is key component is utilizing Azure Blob Storage. We also learned about the other storage options for IaaS VMs, such as Azure Managed Disks. Azure File Share is useful in sharing files.

We learned about Azure Cosmos DB and its database APIs for structured or non-structured databases. Finally, we discussed choosing the right API for your Azure Cosmos DB solution, as well as useful resources and links for more information.

Check Your Knowledge

1. What is the difference between structured and non-structured data?
2. What Azure SQL service (PaaS) supports instance-scoped features like CLR and Service Broker?
3. What free tool can you use to manage your Azure storage locally on your computer?
4. Why would you consider using Azure Cosmos DB for your data?
5. What is the difference between Azure Blob Storage and Azure File Share?

For the answers to these questions, see the [Appendix](#)..

Recommended Learning Resources

“AZ-104: Implement and Manage Storage in Azure.” Microsoft Learn, <https://oreil.ly/FQovz>.

“Azure Cosmos DB.” Microsoft, <https://oreil.ly/1IOdN>.

“Azure Cosmos DB Documentation.” Microsoft Learn, <https://oreil.ly/mqK7O>.

“Azure Cosmos DB Free Tier.” Microsoft Learn, March 2, 2023, <https://oreil.ly/f22im>.

“Azure SQL for Beginners.” Microsoft Learn, <https://oreil.ly/MBKLb>.

“Azure Unblogged – Replace Your File Server with a Serverless Azure File Share!” ITOpsTalk, February 24, 2020, YouTube video, <https://oreil.ly/ILPa9>.

“Databases, Containers, and Items in Azure Cosmos DB.” Microsoft Learn, February 28, 2023, <https://oreil.ly/P3A4z>.

Singh, Vinay. “10 Benefits of Cloud Storage.” Cloud Academy, June 30, 2023, <https://oreil.ly/tJTOY>.

“Time to Live (TTL) in Azure Cosmos DB.” Microsoft Learn, January 10, 2023, <https://oreil.ly/r7ASo>.

“What Is Azure Cosmos DB for MongoDB?” Microsoft Learn, September 13, 2023, <https://oreil.ly/00Lzf>.

¹ Wikipedia, “Microsoft SQL Server,” https://en.wikipedia.org/wiki/Microsoft_SQL_Server

² Peter Bickerton, 2022, “Why cloud computing is important for data-driven bioscience research,” Earlham Institute, <https://www.earlham.ac.uk/articles/why-cloud-computing-important-data-driven-bioscience-research>

³ Microsoft Customer Stories, “Sentara Healthcare improves lives for patients, members, and clinicians by centralizing data in Azure,” <https://customers.microsoft.com/en-us/story/769786-sentara-healthcare-health-provider-azure>

- ⁴ G2.com Business Software Reviews, “20 Cloud Cost Management Statistics to Help Soar and Save,” <https://www.g2.com/articles/cloud-cost-management-statistics>
- ⁵ Mary Shacklett, “10 big data and analytics resolutions for 2022,” TechRepublic.com, <https://www.techrepublic.com/article/10-big-data-and-analytics-resolutions-for-2022>

Part III. Artificial Intelligence, Machine Learning, Big Data, IoT, and Security

This third part of the book and its chapters focus on teaching the fundamental concepts of artificial intelligence, machine learning, Internet of Things (IoT), edge technologies, big data, analytics, cloud security, DevSecOps, and identity and access management in Microsoft Azure.

Chapter 6. Artificial Intelligence, Machine Learning, and Cognitive Services in Azure

Making applications easy to use and enabling users to focus on the work that adds the most value will be integral when developing applications going forward. Artificial Intelligence is a central technology for achieving this. Using Azure, all options are covered: a developer can use Cognitive Services to add cognitive intelligence into applications without needing any AI/ML skills. Machine learning engineers and Data Scientists can use Azure Machine Learning in all parts of the project lifecycle. Citizen Developers using low code/no code solutions such as Power Apps can use AI Builder to make an app more intelligent.

— Håkan Silfvernagel, Manager AI and Big Data at Miles AS, Microsoft AI MVP and Microsoft Certified Trainer

In the previous chapter, we learned about data storage and databases in cloud computing using Azure. We learned about cloud storage options and data types, especially in our applications' processes and operations. In this chapter, we will dive deeper into how we can use the data hosted in the cloud through artificial intelligence (AI), machine learning (ML), and other cognitive services in Azure that will help us build intelligent systems.

Artificial Intelligence on Azure: An Introduction

Before diving deep into the vital key and technical concepts of AI and ML and learning about the cognitive services in Azure, we need to know the basics of what they are and how they are helpful to us, our applications, organizations, and society.

AI is predicted to reshape our work as technology replaces mundane tasks. Some manual tasks need to be performed by human beings; however, we need to automate, digitalize, and modernize how we conduct some tasks. Using AI, we can improve efficiency, effectiveness, and the speed of delivery to our business.

Even though AI technologies have been seen in many science fiction movies and TV shows, we still have much to learn about this technology. The sci-fi movies we watch also give us the unclear and common misconception that AI and machine learning will take away our jobs and that this technology will replace the human workforce. Thus, we must understand the value and significance of this critical area of computer science.

Gartner notes that AI applies advanced analysis and **logic-based techniques**, including machine learning.¹ AI is used to interpret events, support and automate decisions, and take action. It is a technical subject that requires advanced scientific and mathematical skills for those with AI-related roles in the industry. Programming languages used with AI-related jobs include Python, C#, and R. Other tools on many platforms can be used, including the AI tools available in Azure.

A press release, **“Gartner Identifies Three Imperatives Driving the Top Trends in Data and Analytics for 2022”**, noted:

AI is becoming more pervasive, yet most organizations cannot interpret or explain what their models are doing, resulting in a lack of trust and transparency. Organizations are not prepared to manage the risks of fast-moving AI innovation and are inclined to cut corners around model governance including security, escalating the negative consequences of misperforming AI models, such as incorrect business decisions or worse, those impacting life or death.

As regulations for Artificial Intelligence (AI) proliferate globally, they are mandating certain auditable practices that ensure trust, transparency and consumer protection. By 2026, Gartner anticipates organizations that develop trustworthy purpose-driven AI will see over 75% of AI innovations succeed, compared to 40% among those that don't.

This tells us that AI is the future of technology that will expand our horizons and achieve new cognitive services. On the other hand, because AI impacts people's lives and our society, we need to focus on how to handle security, deal with the risks associated with AI, and improve performance outcomes.

AI is being used and implemented in application development, cloud development, data science, and more. The Microsoft Azure AI platform offers a wide range of services and capabilities to build intelligent applications.

Azure OpenAI Service and Evolution of Chat-GPT

At the beginning of 2023, Microsoft announced the availability of Azure OpenAI Service, which enables organizations to build advanced AI models like DALL-E 2, Codex, GPT-3.5, and GPT-4.² The release of Azure OpenAI promises features of enterprise quality and enables the opportunity to create an AI-powered infrastructure on Azure.

If your organization, developers, or AI engineers want to design and build advanced and optimized AI models, the Azure OpenAI Service provides the

opportunity for modern AI solutions at a huge scale. A few examples of AI-enhanced tools and technologies already being used include the AI pair programmer GitHub Copilot and PowerApps and **GPT-4-integrated Microsoft products** like Power BI and Microsoft Designer are enhanced with AI with natural language processing models.

In addition to **Azure OpenAI**, ChatGPT (Generative Pre-trained Transformer) has gained popularity, which also caught the interest of Microsoft, the public, and other tech giants.³ Microsoft is joining its vision and mission to make breakthrough AI technologies by collaborating with OpenAI.⁴ If you are curious about chatting with Chat-GPT, you can check it out at <https://chatgpt.com>.

Finally, it is important to note that although the Azure OpenAI Service is generally available, it is actually not open for public use. For an organization to use this service, it must submit an **application**, which includes the conditions for following the code of conduct for responsible AI.

AI Technology Innovations and Terms You Need to Know

The following are some AI technologies you should know:

Natural language processing (NLP)

NLP is an intuitive communication between humans and intelligent systems using human languages. It guides interactive voice response (IVR) systems by processing language. For example, a typical application of NLP is chatbots on websites.

Internet of Things (IoT) with AI

IoT includes the network of physical objects or devices that have sensors. These sensors are embedded to interact with other devices and collect data telemetries. IoT enables these devices to connect and exchange data. IoT is being used in

Smart Homes to automate and control heating, security systems, lighting, heating, etc. It is also widely used in areas like healthcare for health monitoring and in transportation to improve traffic management.

Computer vision (CV)

CV can process real-life images through image capturing, processing, and analysis. It allows machines to gather contextual, meaningful, and valuable information from the real world. CV techniques have unique requirements for infrastructure and technology that may differ from traditional approaches to ML. CV is getting better at accurately identifying objects for applications, such as self-driving cars, automated retail stock checks, and automated drones.

Synthetic data

In ML terms, synthetic data is artificially generated and resembles or replicates the statistical properties of actual data without the accurate data's identifying properties (e.g., name, date of birth, ID numbers, or other personal details). AI requires an enormous capacity for data. Synthetic data to meet this need can be generated using various algorithms, such as random number generation, simulation, and statistical models. The goal of developing synthetic data is to provide a representative sample of the data distribution, including the relationships between variables, without exposing sensitive data. Benefits of using synthetic data in ML include protection of sensitive information and data consistency for training and testing.

Virtual agents or conversational AI agents

A typical virtual agent is a computer application that interacts with humans. Many websites provide customer

service through chatbot agents to assist and interact with customers to answer their queries. Virtual agents are characterized as software as a service (SaaS).

Edge AI

An AI technique embedded at the touchpoint where physical devices meet the digital world, **edge AI** enables IoT. One example of this is that you can also perform a task such as **deploying AI and machine learning computing on premises and to the edge**.

Generative AI (GenAI)

Generative AI technology learns about artifacts from data; after learning from data, it generates innovative creations that are similar to the original without repeating or copying it. It uses generative models for creating content such as media, images, text, etc.

Because the AI technologies listed are useful across industries and their use is expected to increase, the demand for AI professionals working with these technologies is also growing.

Why Should You Learn AI?

Cloud development and engineering, especially in Azure, involve learning new technologies and concepts. It is essential to understand the different technologies and their applications in real-life cases. Whether you have a role in leadership, sales or development, or engineering, you need to learn at least the fundamental concepts of AI.

AI is an essential modern skill to have

Learning AI is a skill that is required for building IT solutions, especially in the cloud. Based on **O'Reilly's Radar report on the state of AI adoption**, the most significant

barriers are issues with data and a lack of skilled people. The future of work involves automation, and as such, everyone should know the basic concepts of AI. Acquiring such skills will future-proof your career. The rise of AI in the commercial world will also create many job opportunities in various industries. Aside from gaining the necessary skills in AI, there are also **some important considerations to think about before implementing AI into any type of business.**

High demand of AI-related jobs and careers

Gaining knowledge and career experience in this field will help you get job roles such as machine learning engineer, software engineer working with modern applications with AI technologies, data analyst, research engineer, business intelligence developer, and data scientist.

Help build better, intelligent, and inclusive applications

Since AI can help us build better and more intelligent systems, we can contribute to making our lives and those of others easier. Also, the more we develop our applications to be more inclusive and human, the better and more useful our systems become for daily purposes, for example, virtual assistants and customer service chatbots. These AI-built tools and services are helping improve our product delivery and customer service satisfaction, which benefits any business or organization.

AI is a flexible and versatile field

AI improvements and solutions impact many industries like customer service, finance, security, fraud detection, and even healthcare. **virtual health assistants (VHA)** can assist medical patients, for example, ensuring patients are on the correct prescribed medication, recommending essential health treatments, or developing ideal diet recipes based on

the data history of previous illness. If you're interested in learning more, read the medical paper "[The role of artificial intelligence in healthcare: a structured literature review](#)" on the role of AI in healthcare.

Benefits of AI to Businesses

AI is a trending topic for companies, especially in ecommerce.⁵ AI's capabilities are evolving, allowing companies to improve customer engagement in real time, manage their operations, and ensure business continuity, especially when many are working and doing business hybrid and remotely. Organizations are creating new pathways to innovate and expand opportunities using AI.

AI has practical benefits for companies and businesses around the world, including:

Increased efficiency and productivity

AI can help automate manual and repeatable tasks and processes. By implementing AI in these manual processes, employees and teams can focus on more creative and strategic work. AI can also help increase overall efficiency and productivity.

Better decision making

AI can analyze data and provide insights that can advise better business decisions. It can help businesses to collect, process, and analyze large amounts of data. By using advanced algorithms, AI can identify patterns and trends that may not be immediately apparent to humans, and provide insights to help make informed decisions.

Improved customer experience

AI can help companies provide personalized customer experiences, such as customized product or user

recommendations or more efficient and effective customer support.

Cost savings

By automating tasks, AI can help companies reduce labor costs and increase cost savings. AI can be used to optimize the supply chain, by predicting demand, optimizing inventory, and identifying the most cost-effective shipping routes. By improving supply chain efficiency, businesses can reduce costs and increase profits.

Competitive advantage

Businesses that adopt AI-driven technologies can gain a competitive advantage by making better use of data and quickly responding to changing market conditions.

Predictive maintenance

AI can support companies in predicting when equipment or a machine is likely to fail, allowing for proactive maintenance that can reduce downtime and improve overall operational efficiency.

Fraud detection

AI can assist in detecting fraudulent activity more quickly and accurately, reducing financial losses and improving security.

These benefits show that AI can be a helpful tool to use in our applications and systems. However, we need to be responsible in using AI it and study it carefully. Responsible AI in cloud development and machine learning are good topics to study. We will learn more about them throughout this chapter.

NOTE

An infographic report about AI in retail was published by **Juniper Research**. This report projects the total retail spending for machine learning in 2023 and the forecasted annual software spending on AI compared to 2018. Spending on ML in AI is expected to continue to grow at a projected rate of 230% between 2019 and 2023. Approximately 325,000 retailers will be using machine learning by 2023, and intelligent checkout technologies in ecommerce or shops will help facilitate 1.4 billion in transactions in 2023.

Machine Learning

Machine learning (ML) is a subfield of AI that specializes in developing algorithms and statistical learning models. These algorithms and learning models enable computers to learn from and predict data, which involves collecting and feeding a massive amount of data into a learning ML model. The model will help identify patterns and relationships within the data and make predictions based on those patterns. The more accurate the data the model is trained on, the more accurate the predictions.

Three different types of ML are:

Supervised learning

This type of ML involves training a model on labeled data where the correct output is known. The goal is for the model to make predictions on new, unseen data similar to the training data.

Unsupervised learning

This type of ML involves training a model on unlabeled data where the correct output is not known. The model aims to identify patterns and relationships within the data.

Reinforcement learning

This type of ML consists of training a model to make decisions in an environment by receiving rewards or penalties for specific actions.

According to a Gartner study, “**What Are Machine Learning and Deep Learning?**”:

ML is a critical technology that enables AI to solve problems. Despite common misperceptions, machines do not learn. They store and compute—admittedly in increasingly complex ways. It is a purely analytical discipline. It applies mathematical models to data to extract knowledge and find patterns humans would likely miss. ML also recommends actions but does not direct systems to take action without human intervention. Machine learning creates an algorithm or statistical formula (referred to as a “model”) that converts a series of data points into a single result. ML algorithms “learn” through “training,” in which they identify patterns and correlations in data and use them to provide new insights and predictions without being explicitly programmed.

These complex ML system components are composed of machine learning algorithms, training data, evaluation metrics, business logic, and more. Data engineers, analysts, data scientists, ML engineers, and users are also involved in developing ML systems. It has a range of applications, including recognition for speech and languages. It also includes natural language processing.

NOTE

AI and ML are often used interchangeably, but even though they are not the same, they are closely interconnected. In a use case, for example, an AI system is built on techniques like machine learning to create and study the patterns in the training data. Data scientists then optimize these ML models to provide the best results.

MLOps and DevOps: What's the Difference?

In cloud development and software development, we have DevOps. In machine learning, there is also the term *MLOps*, which means **machine learning operations**. O'Reilly's Radar reports provide a good article titled "**MLOps and DevOps: Why Data Makes It Different**" on this topic.

Both MLOps and DevOps aim to streamline the development and deployment of ML models.

DevOps are practices that focus on teamwork and communication between development and operations teams to automate the process of software delivery and infrastructure management. DevOps increases the speed and quality of software delivery and makes the process more efficient and less error-prone.

MLOps extends the principles of DevOps to the field of machine learning. It aims to automate the end-to-end lifecycle of ML models, from development and testing to deployment and management. It also helps ensure that learning models can be deployed and managed in production just as easily as traditional software applications.

The practices of both MLOps and DevOps aim to ensure that learning models are developed in a repeatable and scalable way. These practice combinations also help ensure that they are tested and validated before deployment and can be monitored and updated in production. They also maintain the reliability and performance of machine learning models in production and reduce the risk of errors and failures.

Businesses also need to establish an AI business strategy to identify use cases and success metrics. Typical methods include identifying the benefits of risk reduction, increased speed of tasks, increased number of sales, increased count of satisfied customers, etc.

We're just at the beginning of an explosion of intelligent software.

—Tim O'Reilly, founder, CEO, and chairman of O'Reilly
Media

As **Tim O'Reilly** said, we are only in the initial phase of evolving and growing intelligent software and applications.

Cloud development in Azure and other cloud platforms will be fundamental in how we do business, work with innovations and digital products, and serve our customers in the future.

Deep Learning in ML

As a subset of machine learning, deep learning uses artificial neural networks with multiple layers to model complex relationships and patterns in data.

Deep learning has revolutionized many areas of AI, from computer vision and NLP to playing popular games like chess and Go. The models for this type of learning have been trained to achieve state-of-the-art accuracy on many tasks and keep pushing the boundaries of what is possible with AI.

In addition, the deep learning in ML is used in many real-world applications, such as autonomous vehicles, speech recognition systems, image classification, and even medical diagnoses.

Deep learning is constantly evolving, with new techniques and architectures being developed all the time.

Ethical and Responsible AI on Azure

Important questions are being raised about what humans should do with these systems we teach to think like us and what these learning systems can or should do. These questions include the possible risks in letting them do the work for us and how we can effectively manage and control AI-built systems.⁶

In practical terms, ethical AI makes sure that the AI initiatives of any organization, company, or entity follow and maintain human dignity and do not harm people or society. It is broad in spectrum, which means that any AI technologies and products should be built and designed with fairness,

liability, safety, and security. An example often raised for ethical AI is the use case of self-driving cars and accidents.

Pew Research Center indicates that in as little as a decade from today, ethical AI design will be widely adopted by many. The research showed that many are concerned that AI will continue to be mainly concentrated on optimizing returns and social authority and that stakeholders will need help to reach unanimity about ethics.⁷

When asked whether by 2030 the AI systems used by organizations would employ ethical principles focused primarily on the public good, 68% believed they would not. The research added that “ethical” implies adopting AI in a transparent, responsible, and accountable manner. For others, it means ensuring their use of AI remains consistent with laws, regulations, norms, customer expectations, and organizational values. Ethical AI also would aim to guard against biased data or algorithms, assuring that automated decisions are justified and explainable.

Microsoft has developed **Fairlearn**, an open source suite of tools for **Responsible AI**. Fairlearn is a Python package that implements several algorithms to detect and mitigate group fairness issues in ML models. It is an essential toolkit for ML researchers and developers because, through responsible AI, we want to train ML models accurately and teach development teams to behave reasonably. Fairness in responsible AI must include fairness towards individuals and groups with a way to fix fairness issues in AI.

TIP

Microsoft aims to provide education and spread awareness of responsible AI to everyone. See the resources at **Microsoft’s website**. These resources can help you, your organization, and your team to responsibly use and develop AI for any innovation from concept, through development, to deployment.

The **Responsible AI Toolbox** will show you how to develop and use AI systems more responsibly.

Ethical AI and responsible AI are essential in any design, development, and implementation process.

Azure AI and Cognitive Services

Azure Cognitive Services is part of Azure AI services that helps build intelligent and innovative applications by implementing cognitive intelligence into our applications.

Regardless of framework and programming platform type, your Azure services are usually accessible for development using client library SDKs for different programming languages, REST APIs, and platforms built with user interfaces. Because of the variety and flexibility of these cognitive services and tools, you don't need to be an expert in AI to add and apply cognitive services features to your applications.

Uses for Azure AI Services include being able to customize pre-trained machine learning models using AI innovation and research. We can deploy cognitive services from a cloud computing platform to **edge computing**.

Azure AI Services allow you to build cognitive solutions to see, hear, speak, understand, and make decisions. These solutions are grouped into Speech, Vision, Language, and Decision.

Table 6-1 describes the commonly used Azure AI Services based on different cognitive categories and how they are useful.

Table 6-1. Azure Cognitive Services categories

Cognitive API category	Name	Description
Speech	Speech Service	Add speech-enabled features to applications in different ways, such as speech-to-text, speech translation, text-to-speech, and many more.
Vision	Computer Vision	Provides access to cognitive algorithms for image processing and image recognition. If you want to customize and build your image classifications, there is also the Azure AI Custom Vision cognitive service.
Vision	Face Service	Access different face algorithms that detect and recognize faces.
Language	Language Service	Analyze and understand text; you can use this service with the help of NLP features.
Language	QnA Maker	Build a question-and-answer AI service based on content structure using the QnA maker service .
Language	Language	Language Understanding (LUIS) LUIS is a cloud-based conversational AI that uses ML to help predict the overall meaning of a

Cognitive API category	Name	Description
		person's conversational or natural language text.
Decision	Translator	Translate the messages and text in your application using the cognitive translator service that provides machine-based real-time text translation.
Decision	Anomaly Detector	Detect and monitor possible abnormalities in time series data. The Azure AI Anomaly Detector has an interactive demo that will help you understand how it works. To use this demo, you need an Anomaly resource, endpoint, and API key.
Decision	Content Moderator	Add monitoring for possible offensive, undesirable, and risky content in your application or solutions. Content Moderator moderates content by scanning and flagging images, text, videos, etc. This will help build compliance and regulations for your users.

TIP

Building Intelligent Apps with Cognitive APIs (O'Reilly Media) provides in-depth knowledge about building innovative applications using cognitive APIs.

These cognitive services help make our applications more intelligent and innovative while helping us save time.

NOTE

In July 2022, [Microsoft made Azure Cognitive Services as Limited Access service](#) to the public to follow [Microsoft Responsible AI Principles](#).

Limited Access services require users to register and only customers managed by Microsoft are eligible for access. For more information, check out Microsoft's documentation on [Limited Access features for Cognitive Services](#).

Azure Machine Learning

Azure ML is a service for machine learning-related solutions, which helps in improving and managing an ML project's lifecycle. Data scientists and engineers working with ML or AI can use the available features and tools for designing and developing ML workflow tasks.

Tasks related to this field include designing ML models, training them, and managing the lifecycle of MLOps. An ML model can be created using open source platforms, such as PyTorch, [TensorFlow](#), or scikit-learn. MLOps tools help you monitor, retrain, and redeploy models.

[Figure 6-1](#) shows how machine learning models are developed. As with many processes, you must first define the task. Then explore and prepare your data for a specific purpose or use case. At this point, you'll be ready to train and validate your model, then deploy it.

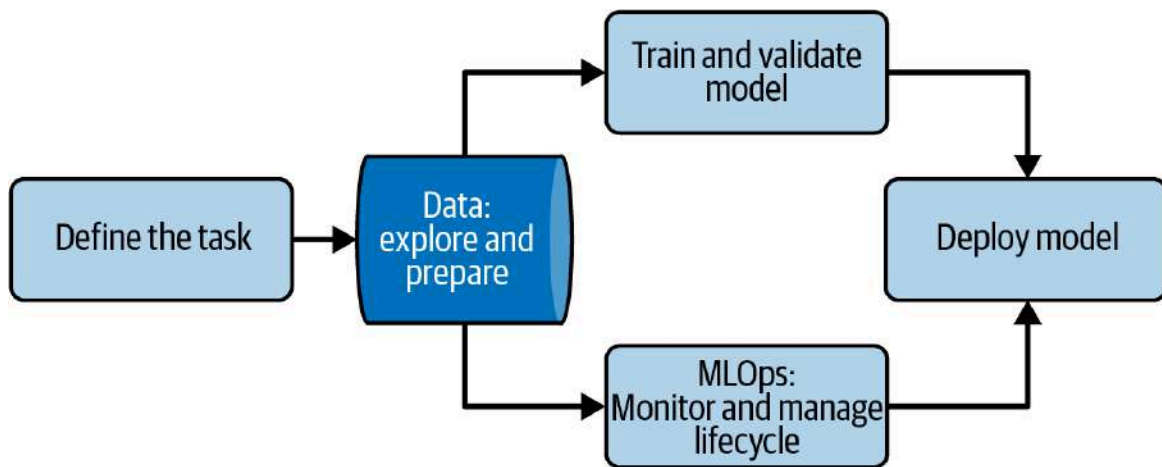


Figure 6-1. Azure Machine Learning's development lifecycle

Once your model is deployed, you'll want to monitor and manage it, ultimately bringing you back to exploring your data and creating a continuous loop. Some workflows push new data every few hours, others every few days. Your MLOps team will determine your needs as they monitor and manage the lifecycle.

The development lifecycle for an ML project is often repetitive and iterative. It requires experimentation and analysis of data, machine learning algorithms, and training models.

Like any other project in software engineering and cloud development, Azure Machine Learning cloud service enables its users to collaborate on a machine learning project. In an ML project, you need to create an **Azure Machine Learning workspace** as a logical container for the project.

A ML workspace in Azure is a central hub that allows you to manage and collaborate on all your ML experiments, models, and assets. It provides a comprehensive set of tools and resources for data scientists, ML engineers, and developers to work together and develop high-quality ML models.

By utilizing an ML workspace in Azure, you can:

- Collect, store and organize your data
- Develop and train ML models
- Track your ML experiments and results

- Deploy ML models to production
- Monitor and manage the health of your deployed models

You can also integrate it with other services and tools, such as Azure Databricks, Azure Data Factory, and Azure DevOps, to provide a seamless and collaborative environment for ML development and deployment.

Machine Learning Studio

Azure ML Studio is the web portal for data scientist developers. It is a cloud-based IDE that is part of the Azure AI platform. It provides a user-friendly graphical interface that enables data scientists and ML practitioners to build, deploy, and manage ML models without having to write code.

ML Studio is Microsoft's central point of contact for machine learning-related computation. This suite will replace the Machine Learning Studio Classic, which will retire in 2024. (To prepare for this event, please see this helpful guide for the **Azure Machine Learning Adoption Framework** for a basic “lift and shift” migration to the new system.)

Uses of Azure ML Studio include:

Building machine learning models

Azure ML Studio provides a wide range of pre-built algorithms that can be used to build models for tasks such as classification, regression, and clustering.

Data preparation

The studio provides tools for data preparation and conversion, allowing data scientists to quickly clean, process, and format their data before building models.

Model deployment

Once the models are built, the tool provides possibilities for deploying the models as web services that other applications

and users can consume.

Model management

Provides a central location to manage and monitor machine learning models, allowing data scientists to track their models and their performance over time.

Collaboration

The studio enables sharing of models and workflows, which helps data scientists work together on ML projects and share their work with others.

Menu options in Azure ML Studio are divided into three main sections:

Author

Deals with the creation of the code and set up of your machine learning processes

Assets

The resources created and stored within the Author section. Assets include the pipelines created in the designer that controls the resources' workflow, from inputting datasets over a pipeline to the endpoints for the output (e.g., connection to real systems via REST API).

Manage

The system behind the scenes including computation clusters and instances, data stores where the datasets are stored, and integrations into other systems.

Figure 6-2 illustrates creating an Azure Machine Learning workspace to get started creating an ML project.

Azure ML Studio is a valuable tool for data scientists and machine learning practitioners looking to build, deploy, and manage ML models in the cloud.

[Home](#) > [Azure Machine Learning registries](#) >

Azure Machine Learning

Create a machine learning workspace

Basics

Networking

Advanced

Tags

Review + create

Resource details

Every workspace must be assigned to an Azure subscription, which is where billing happens. You use resource groups like folders to organize and manage resources, including the workspace you're about to create.
[Learn more about Azure resource groups](#)

Subscription *

Azure Subscription Account

Resource group *

(New) rg-learningazureoreillybook

[Create new](#)

Workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Workspace name *

ws-learningazuremachinelearning

Region *

North Europe

Storage account *

(new) storagelearningazurebook

[Create new](#)

Key vault *

(new) kv-learningazurebook

[Create new](#)

Application insights *

(new) appinsights-learningazurebook

[Create new](#)

Container registry *

None

[Create new](#)

Review + create

< Previous

Next : Networking

Figure 6-2. Azure Machine Learning workspace created on Azure Portal

Automated Machine Learning (AutoML)

AutoML is perfect for creating models without using code. It's a no-code UI, meaning you can work with ML projects without any programming

skills. There are **data featurization options** in AutoML that are worth checking out. These help automate ML experiments.

AI Builder for Power Platform

AI Builder is a feature in the Power Platform that provides an easy-to-use, no-code platform for building and deploying custom AI models for specific business problems. It is designed to help organizations leverage the potential of AI and ML to automate processes, make predictions, and improve decision making without requiring specialized data science or AI skills.

With AI Builder, users can select from a range of pre-built templates and models optimized for common business scenarios or create their custom models from scratch. The platform provides a simple, intuitive interface for defining input data and desired outcomes and training, testing, and refining models.

Once an AI model is built, it can be easily integrated into Power Apps, Power Automate, and other Power Platform components, enabling organizations to quickly and easily apply the insights generated by the model to their business processes.

You can use AI Builder in different ways depending on the model you use. You can use AI models in the formula bar, add components using AI Builder, and use **Microsoft Power Fx** expressions to consume AI Builder models in Power Apps.

Power Fx is a low-code, general-purpose language designed and used on spreadsheet-type formulas. It is a declarative, strongly typed, and functional programming language, with imperative logic and state control available.

For a more detailed look, see the learning paths and modules in the Microsoft documentation **AI Builder on Power Platform**.

Azure Applied AI Services

Applied AI Services is also part of the Azure AI platform that offer a range of tools to support the entire AI lifecycle, from data preparation and model training to deployment and management. It combines Azure Cognitive Services, task-specific AI, and business logic to offer turnkey AI services for standard business processes.

You can automate document processing, improve customer service, understand the root cause of anomalies, and extract insights from any content with Azure Applied AI Services and extend it using your AI models from Azure ML. Specialized AI services for specific business scenarios, like modernizing business processes with task-specific AI, enables you to accelerate development with built-in business logic to launch solutions in days rather than months.

Applied AI in Azure is a suite of artificial intelligence and machine learning services provided by Microsoft Azure to help developers build and deploy AI solutions. Here are some of the Applied AI Services and how they can be valuable for developers:

Cognitive Services

These pre-built APIs allow developers to add features such as NLP, computer vision, and speech recognition to their applications with minimal code. Developers can quickly add AI capabilities to their projects without building models from scratch.

Cognitive Services in Azure follow guidelines and share information about responsible use of AI in applications.

Check out [Microsoft's guide for responsible AI with Cognitive Services](#) in different areas.

Azure IoT Edge

IoT Edge is a platform for running AI models on IoT devices, allowing developers to analyze data at the edge and make

decisions in real time. This can be useful for developers working on IoT projects that need to run their models on resource-constrained devices.

Databricks

Databricks in Azure is an Apache Spark-based analytics platform providing a collaborative environment for building and running large-scale ML models. It offers a fast, scalable way for developers to process large amounts of data and build models with high accuracy.

Form Recognizer

Form Recognizer uses AI and ML algorithms to extract data from structured and semi-structured records such as invoices, receipts, and contracts. This tool allows its users to automate the data extraction process from these records, which helps minimize manual data entry. It also improves the accuracy of data. Customized training is also supported in this service.

Cognitive Search

This service can build AI- and ML-powered search experiences within applications. Common use cases for Cognitive Search services in AI and ML include text analytics, image analysis, NLP, etc. Azure Cognitive Search provides a robust, adaptable platform for building AI- and ML-powered search experiences, whether you are a data scientist or a full-stack developer.

Azure AI Bot Service

Azure AI Bot Service is a platform for creating, deploying, and operating intelligent conversational bots in Azure. It delivers an integrated environment for developers to build

and deploy bots in various channels, such as websites, messaging platforms, Teams, Skype, etc., on the cloud. For developers working in AI and ML, the Azure AI Bot Service provides a convenient and powerful platform to build and deploy conversational bots powered by AI and ML algorithms. The platform offers several built-in features, such as NLP and language understanding (LUIS), that can be leveraged to build sophisticated conversational bots that can understand and respond to user requests in a human-like manner.

Azure AI Immersive Reader

Azure AI Immersive Reader is a tool for reading and understanding text. You can use these features to develop solutions for converting text content to speech, highlighting text line-by-line, picture dictionaries, and more. The main goal of Azure AI Immersive Reader (see the [Microsoft documentation](#)) is to support readers or users with disabilities in reading and comprehension. In addition, it can help individuals of all ages and abilities improve their reading skills and increase their understanding of written text.

Azure Metrics Advisor

This tool helps protect organizations' growth by enabling them to make the right decision based on intelligence from metrics of businesses, services, and physical assets. Azure Metrics Advisor uses AI to perform data monitoring and anomaly detection in time series data, to provide insights on system health and performance. The service automates the process of applying models to data and provides a set of APIs and a web-based workspace for data ingestion, anomaly detection, and diagnostics, without needing to know ML. Developers can build AIOps, predictive maintenance, and

business monitoring applications on top of the service. Furthermore, it uses AI and ML to deliver anomaly detection and advice capabilities. Developers can take advantage of it by leveraging pre-built APIs and tools to help them build and deploy AI solutions quicker and more efficiently.

From a broad perspective, using AI and ML in Azure AI Metrics Advisor optimizes system monitoring and performance, enabling organizations to recognize and fix issues before they evolve into significant problems.

TIP

To learn more about Azure AI, see *Azure AI Services at Scale for Cloud, Mobile, and Edge* by Simon Bisson, Mary Branscombe, Chris Hoder, and Anand Raman (O'Reilly).

The Azure AI infrastructure features **high-performance computing (HPC)** that uses a large number of GPU-based computers to solve complex mathematical tasks, including those in engineering, weather modeling, finance, genomics, and simulations. In addition, **a software as a service (SaaS) solution uses HPC** to solve problems in computer-aided engineering on Azure's platform.

Learn By Doing (Try It!)

The following are recommended tutorials in Microsoft's documentation:

- **Quickstart: Create workspace resources you need to get started with Azure Machine Learning**
- **Tutorial: Train an image classification TensorFlow model using the Azure Machine Learning Visual Studio Code Extension**
- **Tutorial: Forecast demand with no-code automated machine learning in the Azure Machine Learning studio**

- [Quickstart: Get started with Azure Machine Learning](#)
- [Quickstart: Azure AI Vision v3.2 GA Read](#)
- [Azure Machine Learning hands-on labs](#)
- [Set up no-code AutoML training for tabular data with the studio UI](#)

Summary

This chapter explores the basics of AI and how it relates to ML. Through AI and ML, we can train models using our training data to build functional AI solutions for our businesses and applications.

As a cloud platform, Microsoft Azure provides a wide range of services to automate creating and deploying ML models. The Azure AI services comprises several significant categories: Azure OpenAI, Azure Applied AI Services, Azure AI, Azure Cognitive Services, Azure AI Infrastructure, and Azure Machine Learning. Cognitive Services provides a set of APIs that developers, data scientists, data engineers, and AI experts in different business application areas and categories can use.

We also learned about the many benefits of AI and ML and the importance of developing AI-driven solutions responsibly by using Responsible AI. Microsoft Azure, as a cloud platform is also conscious of and responsible for using AI by implementing Microsoft Responsible AI standards that help ensure that our AI solutions are safe, secure, trustworthy, and ethical.

Check Your Knowledge

1. What is artificial intelligence?
2. What is the name of an open source AI fairness project made with Python and developed by Microsoft?
3. What is the difference between AI and machine learning?

4. List a few of the Azure Cognitive Services that help build and add cognitive features to applications with the help of AI and machine learning.
5. Why do you think responsible AI and ethical AI are important in building AI solutions?

For the answers to these questions, see the [Appendix](#).

Recommended Learning Resources

“AI42.” Meetup.com community, <https://oreil.ly/El6bi>.

“Azure Machine Learning Best Practices for Enterprise Security.” Microsoft Learn, October 18, 2023, <https://oreil.ly/vzf19>.

Crampton, Natasha. “Microsoft’s Framework for Building AI Systems Responsibly.” Microsoft on the Issues, June 21, 2022, <https://oreil.ly/F1k8k>.

Loukides, Mike. “AI Adoption in the Enterprise 2021.” O’Reilly Radar, April 19, 2021, <https://oreil.ly/mf-nJ>.

Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. Sebastopol, CA: O’Reilly Media, 2019.

Gonzalez Sanchez, Adrian. *Azure OpenAI for Cloud Native Applications*. Sebastopol, CA: O’Reilly Media, 2024.

Hall, Patrick, Navdeep Gill, and Benjamin Cox. *Responsible Machine Learning*. Sebastopol, CA: O’Reilly Media, 2020.

Hawk, Jessica. “Technical Leaders Agree: AI Is Now a Necessity to Compete.” Azure Blog, March 8, 2022, <https://oreil.ly/6v6KT>.

“Improve Fairness of Online Systems.” Fairlearn, <https://oreil.ly/0j2Rx>.

Körner, Christopher, and Marcel Alsdorf. *Mastering Azure Machine Learning (Second Edition)*. Birmingham, UK: Packt Publishing, 2022.

Kuznetsov, Gary. “Artificial Intelligence vs. Machine Learning: A Comparison + Interactions & Examples.” Digital Silk, December 21, 2019,

<https://oreil.ly/cK-84>.

Moroney, Laurence. *AI and Machine Learning for Coders*. Sebastopol, CA: O'Reilly Media, 2020.

“Principles and Approach.” Microsoft AI, <https://oreil.ly/V2wxF>.

“Tools and Practices.” Microsoft AI, <https://oreil.ly/4FgNi>.

Serpa, Ygor. “AI Papers to Read in 2022.” Medium, March 4, 2022, <https://oreil.ly/x4Ua8>.

Smith, Chris, Brian McGuire, Ting Huang, et al. “The History of Artificial Intelligence.” University of Washington, December 2006, <https://oreil.ly/4jet7>.

“Welcome to the Elements of AI Free Online Course!” Elements of AI, <https://oreil.ly/JBZLh>.

“Zero to Hero in 4 Weeks with Azure AI.” Microsoft Azure presentation, <https://oreil.ly/cC7mD>.

Zheng, Daniel, Nestor Maslej, Erik Brynjolfsson, et al. “The AI Index 2022 Annual Report.” AI Index Steering Committee, Stanford Institute for Human-Centered AI, Stanford University, March 2022, <https://oreil.ly/MpJmH>.

¹ Gartner, “Artificial Intelligence (AI),” <https://www.gartner.com/en/information-technology/glossary/artificial-intelligence>.

² Microsoft Azure Blog, January 16, 2023, “General availability of Azure OpenAI Service expands access to large, advanced AI models with added enterprise benefits,” <https://azure.microsoft.com/en-us/blog/general-availability-of-azure-openai-service-expands-access-to-large-advanced-ai-models-with-added-enterprise-benefits>

³ Wikipedia.org, “ChatGPT,” <https://en.wikipedia.org/wiki/ChatGPT>

⁴ Microsoft News Center, “OpenAI forms exclusive computing partnership with Microsoft to build new Azure AI supercomputing technologies,” <https://news.microsoft.com/2019/07/22/openai-forms-exclusive-computing-partnership-with-microsoft-to-build-new-azure-ai-supercomputing-technologies>

- 5 Shivbhadrasinh Gohil, “Artificial Intelligence: Growth Boosting Factor in the New Normal,” IoTForAll.com,, <https://www.iotforall.com/artificial-intelligence-growth-boosting-factor-in-the-new-normal>
- 6 Stanford Encyclopedia of Philosophy, “Ethics of Artificial Intelligence and Robotics,” <https://plato.stanford.edu/entries/ethics-ai>
- 7 Lee Rainie, Janna Anderson, and Emily A. Vogels, “Experts Doubt Ethical AI Design Will Be Broadly Adopted as the Norm Within the Next Decade,” Pew Research Center, June 16, 2021. <https://www.pewresearch.org/internet/2021/06/16/experts-doubt-ethical-ai-design-will-be-broadly-adopted-as-the-norm-within-the-next-decade>

Chapter 7. Big Data, Reporting, and Analytics Services in Azure

From the global economic crisis caused by the pandemic to the rapid adoption of AI tools like ChatGPT, businesses have never been more reliant on data and analytics to compete and stay ahead. To make this happen, legacy silos between data analysts versus scientists or archival versus streaming data must be torn down in favor of an interoperable, governed platform for analytics and reporting. Done well, analytics means getting the right information at the right time to the right people—to do this, the right architecture matters.

—George Mount, Founder of Stringfest Analytics,
Microsoft MVP, and author of *Advancing into Analytics*
(O'Reilly Media)

In [Chapter 6](#), we learned about how artificial intelligence (AI), machine learning (ML), and Azure AI Services can be used to build intelligent applications. We explored using these AI, OpenAI, and cognitive services to collect data to develop applications and enterprise-level solutions.

In this chapter, we'll shift gears and dive deeper into how to use and visualize data collected using cloud services to analyze and understand it better.

Big Data, Reporting, and Analytics Services in Azure

Organizations constantly collect, process, and use data in applications on all sorts of devices. Looking at this data at scale through extensive data analysis turns datasets into actionable information. This level of analysis is referred to as big data analytics, and it's the foundation of many data-driven business and enterprise decisions.

Being data-driven as a business or an organization avoids guessing and misconceptions. Visually representing this data makes the information accessible for a wide variety of users. But before we can transfer data into insights, we need a process for data analysis.

Each project merits a different approach. Consider factors like how you would prefer to collect and visualize your data. Where would you like to collect this information, and how would your organization want to manage it effectively? How would you like to optimize this enterprise data to improve services to your users?

Often a **cloud-based data warehouse** has a dedicated analysis service. You can combine your big data solutions on the managed services with private clouds or set up your hybrid operation.

What Is Big Data?

What does working with big data mean? Does it help improve how we handle data collection, reporting, and analysis in our enterprise applications on the cloud or on premises?

Use cases of big data solutions include social media platforms, the IoT solutions, gaming industry, etc. Millions of users generate huge volumes of data through data streams that need to be handled quickly and in real time. Systems built to handle big data are specialized to solve the challenges of the four Vs:

Volume

Technology platforms used in industries like ecommerce, social media, IoT, gaming, etc., continuously generate large sets of data around the clock. For example, **connected cars** across the globe send live telemetry data to their car manufacturing company's systems. This massive volume of data can only be handled by big data systems.

Variety

Data we are familiar with and used to working on is structured data in different types or formats (for example, text, time, number), which we collect and store using relational database systems. Increasingly, the evolving demand and use of unstructured data for things like live streaming of media files, images, feeds on X (formerly known as Twitter), websites, live logs, etc., makes it challenging for data analytics systems. This data variation is one of the challenges big data systems solve.

Velocity

Data velocity refers to the speed of incoming data processing. As mentioned earlier, gaming or IoT systems that require speed of delivery and data processing are good examples of the need for velocity.

Veracity

The veracity or integrity of data is the fourth essential aspect of big data because it checks and verifies data authenticity, trustworthiness, precision, and accuracy. Veracity is vital when the data to be analyzed will be used to make critical analytical decisions, which are crucial for trustworthy output or results.

These challenges can be solved in a cost-efficient way using big data systems. Today's big data technology platforms use distributed processing for scalability, fault tolerance, and high throughputs.

In distributed processing, multiple computing resources work together to execute a task. **Hadoop** is a big data solution built on distributed processing concepts. It can handle computer nodes that can work with up to tens of petabytes (PBs)¹ of data to perform a query task.

For example, Azure HDInsight (a managed Hadoop cloud service) and Azure Databricks are widely used big data platforms on Microsoft Azure. We will learn more about the different big data services and solutions throughout the rest of this chapter.

Figure 7-1 shows the typical data flow process, which involves data collection, storage management, then big data analytics, where a decision-making unit is critical in moving to the network optimization unit.

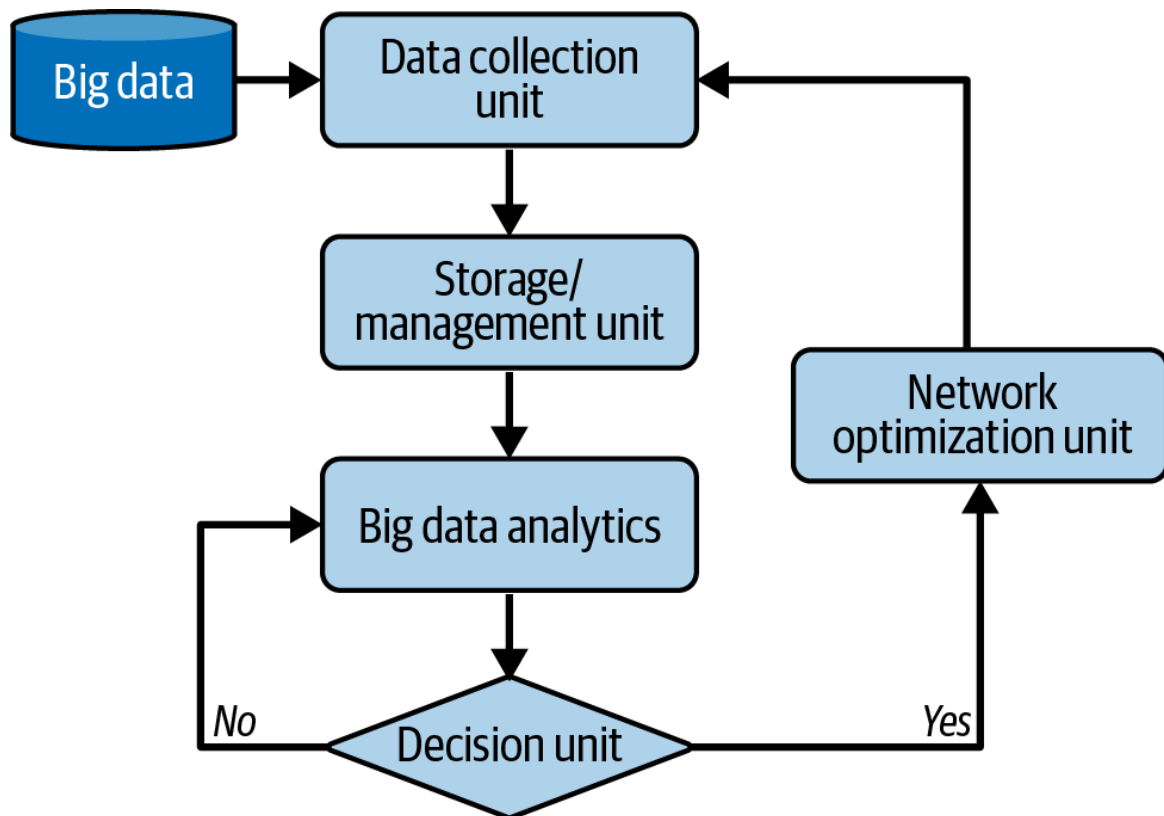


Figure 7-1. Big data flow process

Big Data Solutions in Azure

Cloud services for AI, big data, and analytics are part of the cloud platform Azure provides to its users. The **big data solutions** and services in Azure provide flexibility to users who want the synergy of combining cloud computing capabilities with analytics.

Whether the data is structured or unstructured data or of high volume that requires colossal storage capacity, Azure big data services come with fully managed infrastructure and real-time analytics. In addition, they can be well integrated with other cloud resources in Azure, for example, other analytics, ML, storage, and database services.

Aside from computing, networking, hybrid, and other cloud solutions, Microsoft Azure also provides a wide range of services that can help you set up a significant data infrastructure, including databases, data processing or management tools, analytics, etc., to manage complex data sources.

Big Data Use Cases in Azure

Aside from computing, networking, hybrid, and other cloud solutions, Azure provides a range of services that can help you set up the infrastructure for big data. These include databases for data processing and analytics for machine learning and integrating complex data sources.

The following are use cases of big data in Azure:

Databases

Options for database management include Table Storage, self-managed databases hosted on virtual machines, and other managed databases. The database types available include SQL Server, PostgreSQL, MySQL, and MariaDB. There is also an option for self-managed Table Storage. You can use Azure Cosmos DB for a fully managed distributed database. Cosmos is a globally deployable and replicable service that is scalable, adaptable, and low latency. It also supports several different database engines, and its application programming interfaces are interoperable with a wide variety of tools. Some of these include Cassandra, SQL, Apache Spark, MongoDB, Jupyter, and Gremlin. Azure also includes SQL Data Warehouse and Azure Data Lake, both designed for large-scale structured data, and Azure Data Lake, which is designed for NoSQL types of data.

Analytics

Azure delivers an extensive variety of data analysis products and services. HDInsight and Azure Analysis Services are two examples. They provide a business-grade analysis engine that can aggregate data from a variety of sources and transform it into a semantic business intelligence model that can be accessed with ease. The services can generate interactive dashboards and reports in addition to integrating previously defined data models.

Data engineering

Two primary Azure services create complex data pipelines: Azure Data Factory and Azure Data Catalog. Data Factory provides serverless integration for local and cloud-based data repositories. Azure Data Catalog is a central repository for big data that can be used to manage enterprise data sources. Data analysts, developers, and data scientists can discover, understand, and consume data sources in their data resources and environments. These data engineering tools will be discussed in more detail in the next section.

Machine learning

Azure provides various solutions for artificial intelligence and machine learning, including **Azure Machine Learning Services (AMLS)**. AMLS lets you create customized models for machine learning without requiring programming skills by using a zero-code, drag-and-drop interface and a code-first environment. It is compatible with open source tools and platforms such as **PyTorch**, **TensorFlow**, **Open Neural Network Exchange (ONNX)**, and **sci-kit learn** in Python. AMLS helps automate ML with tools like automated feature selection, algorithm selection, and hyperparameter scanning.

Complex Big Data Pipeline Tools in Azure

Big data can be complex to visualize and analyze. Azure lets you create data pipelines to handle this using Azure Data Factory and Azure Data Catalog.

Azure Data Factory

Azure Data Factory (ADF) allows serverless integration for both cloud-based and local data repositories. Using Data Factory, you can perform extract, load, and transform (ELT) tasks using the available data connectors provided natively by Azure. Some built-in connectors in Data Factory for data sources are from Google Big Query, Amazon S3, and other platforms. The enterprise-ready ADF connections help organizations with visualizing, analyzing, and integrating complex big data use cases. This topic will be discussed further in the upcoming sections of this chapter.

Furthermore, you can transfer or copy existing data from Azure File Storage to Data Factory. This service in Azure allows you to build ETL through a visual editor, which gives flexibility to those who prefer to do it without writing code or scripts. This tool for Data Factory enables you to automate the process with scheduling and drag-and-drop components; you can even create event-based triggers.

Azure Monitor can easily be integrated with Data Factory to help you gain the overall view and visibility to manage the performance of data flowing through CI/CD pipelines. If you'd like to learn more about the architecture of Azure Data Factory, I recommend checking out this [high-resolution infographic](#) by SketchTheDocs.

Azure Data Catalog

Data Catalog is a fully managed metadata catalog for enterprise use. It supports a record for metadata that helps in data asset discovery. It allows you to crowdsource metadata and annotations to users like data analysts, engineers, or data scientists.

It also provides a way to enable these types of users to discover, understand, register, enrich, and consume data sources. Data professionals can share their knowledge and collaborate, which can help make data more searchable and accessible.

You can handle complex big data pipelines using Data Factory and Data Catalog. However, note that Azure is replacing Azure Data Catalog with the Microsoft Purview service by 2025. If your organization has existing Azure Data Catalog accounts, you need to start planning your migration to Microsoft Purview.

Building, Configuring, and Deploying Big Data on Azure

It is recommended to follow a four-step process for designing and developing a new big data solution: evaluation, architecture, configuration, and production.

Evaluation of a big data goal and solution

The first step is to evaluate big data purposes and goals. If you are an organization that is new to this journey, you need to evaluate and understand what kind of data you want to include in your planned solution and why you want it.

For example, scraping data from the web differs from the data telemetry gathered from an IoT device. The amount and type of data used will assist in planning for data ingestion and identifying what kind of storage is required; once you know what type of data needs to be processed, a decision on how to analyze it must be made.

If an organization needs a dedicated big data scientist, then they can use available big data services on Azure to help them get started. Another good option is to use machine learning within the system.

Moreover, if your organization is just starting your cloud journey, you should familiarize yourself with the full spectrum of strategies needed for an Azure migration. Consider starting your project by initially migrating the core applications and processes to the cloud.

Identifying big data architecture

Big data architecture can be classified into different categories based on factors such as data sources, batch processing, orchestration, and real-time streaming requirements and storage mechanisms. Common ways to classify big data architecture include:

Data volume

The size of the data: small, medium, and large. The architecture used for each type of data volume varies based on the amount of data.

Data velocity

The speed at which the data is generated and processed: batch processing, near-real-time processing, and real-time processing.

Data variety

The variety of data types processed: structured data, semi-structured data, and unstructured data. This data variety can be sorted if it was generated by AI, machines, or humans.

Data sources

Data sources can be internal or external.

Processing requirements

The processing requirements for data can be simple or complex.

Storage mechanisms

There are three types of data storage mechanisms: distributed file systems, NoSQL databases, and **Hadoop Distributed File Systems (HDFS)**.

Deployment model

The architecture can be classified based on the deployment model used, for example, on premises, cloud-based, or hybrid.

Considering these different factors makes it easier to classify big data architecture and thus determine your needs. Big data solutions usually involve more than one workload type, for example, batch processing of big data sources at rest, real-time processing of big data in motion, interactive exploration of big data, ML, and predictive analytics.

Therefore, a typical big data architecture should be able to handle complex or large amounts of data for ingestion, processing, and analysis that traditional database systems and applications cannot manage. The **big data architecture style** can help you and your organization gain insights as you plan, design, implement, and build your solutions for big data.

Preparation of Production Environment

After choosing the services you need, you can configure and prepare your production environment. Your exact configuration will depend on your selected services, the data sources, and the environment type.

Based on your big data structure, you typically monitor the processes as often as possible in order to get the best performance and return on your investment (ROI). Azure Monitor and Log Analytics in Azure help many organizations prepare and monitor their workloads, resources, and data for production use.

By monitoring the workloads and applications, creating privacy and security policies, and implementing disaster recovery for your big data system, you are fully utilizing the capabilities you need to be successful in your big data solutions. Based on these processes and guidance in building a big data solution, you should be able to make a solution ready to use in the cloud.

Data Analytics

Data analytics is data collection, extraction, transformation, and modeling. The objective is to collect and identify useful information to assist with business decision making. The origins of data analytics can be traced to the twentieth century when businesses began collecting data to enhance their operations, assisted by advanced in computers and the internet.

Due to the growing significance of data in business, data analytics has become a crucial field of study. It enables businesses to make informed decisions by identifying patterns, trends, and insights through data analysis, then using this information to make predictions.

Today, analytics is an essential component of many enterprises and organizations, including those in healthcare, education, and government, and has become indispensable for businesses and organizations seeking a competitive edge

Most companies are likely already using some analytics solution. Here are the different types of data analytics that organizations most often use:

Predictive data analytics

This is the most utilized type of data analytics. It employs statistical models and ML algorithms to analyze past data and predict future outcomes. It assists organizations in anticipating future trends, causes, correlations, and behaviors and making decisions based on these predictions. Some separate this category into predictive and statistical modeling, but it is essential to understand that both go hand in hand. For instance, an advertising campaign for t-shirts could use predictive data to determine the degree to which the conversion rate correlates with the target audience's geographic location, income bracket, and interests. Then, predictive modeling could be utilized to analyze the statistics for two or more target groups and provide revenue estimates for each demographic.

Prescriptive data analytics

AI and big data combine in prescriptive data analytics to help predict outcomes and identify actions to take. This type of data analytics is composed of random testing and optimization. Using advanced ML features, prescriptive data analytics can help answer questions such as the best possible solution or action. With the help of ML, you have the option to test and verify the correct variables and recommend new ones more likely to generate a positive outcome.

Diagnostic data analytics

Diagnostic data analytics may not be as exciting as predicting the future; however, analyzing past data can be essential in guiding your business. Diagnostic data analytics examine data to understand the cause of an event: why something happened. Techniques such as drill down, data mining, correlations, and data discovery are often employed.

Descriptive data analytics

Descriptive analytics is the backbone of reporting. **Business intelligence (BI) tools** would not be possible without descriptive data analytics, which answer the essential questions like “how many, when, where, and what” of any data. Two subcategories of descriptive data analytics are **canned reports and ad hoc reports**.

Implementing data analytics strategies in companies or organizations can help reduce costs by learning and identifying possible and valuable ways of doing business. Data analytics help organizations make better business decisions by analyzing user trends, customer satisfaction data, etc., which helps create better services and products.

Azure provides a combination of analytics services, including HDInsight and Analysis Services, for enterprise-grade analysis. These services help manage collected data and can also convert these collected data to a usable semantic model for the BI model. You can also use these services to integrate with your predefined database models and create customized reports and dashboards.

Azure HDInsight uses popular open source technologies including Apache Hadoop, Spark, Hive, HBase, and Storm. Its highly scalable platform can quickly scale up or down based on your processing requirements. This means you can process large volumes of data without worrying about infrastructure constraints.

It is a cost-effective solution for big data analytics since you only pay for the processing power and storage you need, and you don't need to worry about the underlying infrastructure costs. Another good feature of this service is that it gives you the flexibility to design and develop solutions that require multiple open source big data technologies supporting a variety of tools, frameworks, and programming languages.

Azure HDInsight delivers enterprise-grade security components such as authentication, authorization, and data encryption. These features and capabilities help to safeguard your data and strengthen compliance with your organization's regulatory requirements.

Azure Big Data and Analytics Services

Azure offers a suite of cloud services for big data solutions and development that help organizations with their operations and explore large volumes of data.

Figure 7-2 shows a big data architecture designed to help handle the complex data of traditional database systems and applications. The big data architecture allows for data ingestion, processing, and analysis. The recommended architecture for big data comprises several components like

data sources, data storage, real-time ingestion, batch processing, and stream processing with ML to reporting.

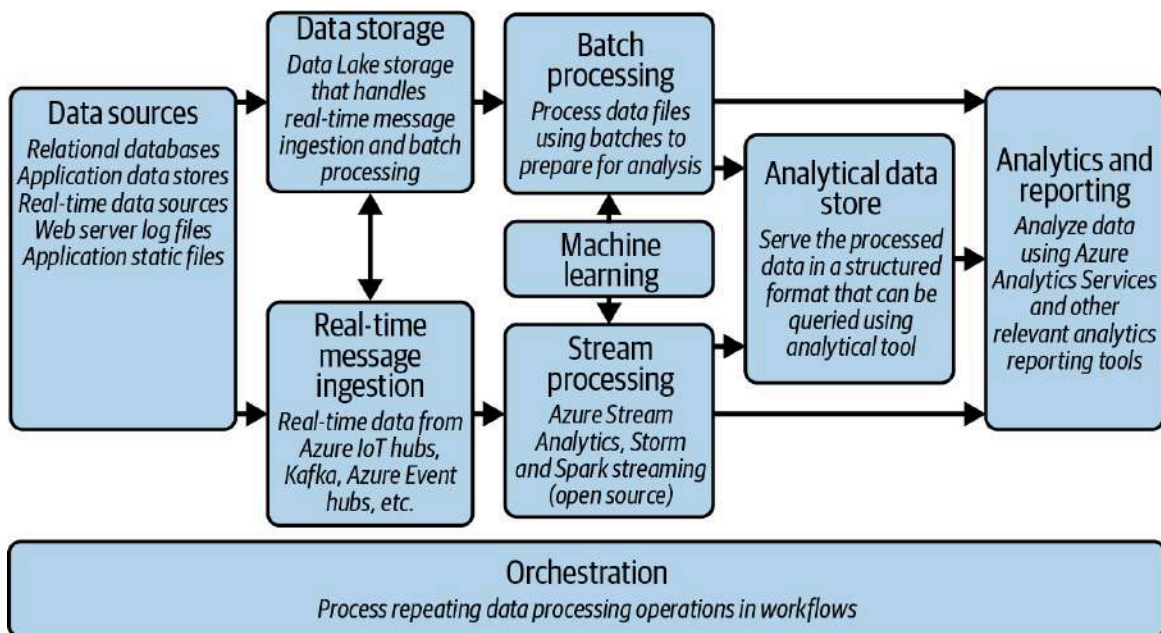


Figure 7-2. Big data architecture style and flow with Azure

Azure includes two types of benefits:

- Managed services including the Azure Data Lake Store, Azure Data Lake Analytics, Azure Synapse Analytics, Azure Stream Analytics, Azure Event Hub, Azure IoT Hub, and Azure Data Factory
- Open source technologies based on platforms like Apache, Hadoop, HBase, Hive, Pig, Spark, Storm, Oozie, Sqoop, and Kafka. These technologies are available on Azure in the Azure HDInsight service.

Azure Data Lake

Azure Data Lake is a scalable and secure cloud-based data storage service that lets organizations store and process big data in its native format. It provides enterprise-grade security features such as encryption, authentication, and authorization and integrates with popular big data tools like Hadoop and Spark.

Azure NoSQL for Big Data and Analytics

NoSQL databases are highly scalable and provide integration support to various applications and workloads. They are popular alternatives to traditional databases and receive robust support from cloud vendors like Azure.

Big data storage for NoSQL databases can use [Azure Cosmos DB](#) and [HBase on HDInsights](#). Additionally, there is also a fully managed data exploration service like [Azure Data Explorer](#) that can be used for analytical databases.

Azure Stream Analytics

Stream Analytics is a real-time stream processing service in Azure that allows organizations to analyze and process streaming data from various sources. This real-time data can be streamed from IoT devices, social media channels, and applications. Users can get real-time insights to help make better decisions. Azure Stream Analytics has built-in machine learning that helps in real-time [anomaly detection](#).

[Data streams](#) are used in data analytics and data engineering to enable sensors, applications, IoT, gateways, and smart monitoring devices to gather and broadcast real-time continuous event data. Streaming data is high volume and has a lighter payload than nonstreaming systems.

Data engineers use Azure Stream Analytics to process streaming data and respond to data anomalies in real time. You can use [Stream Analytics for IoT](#) monitoring, web logs, and point of sale (POS) systems. Developers who work with Azure Stream Analytics can use their preferred IDE, such as Visual Studio Code, which allows them to stream live data using other services such as Azure IoT Hub, Azure Event Hubs, Blob Storage, and more.

Stream Analytics is useful in scenarios where your organization must respond to real-time data events or analyze large batches of data in a continuous time-bound stream. Or your organization can choose whether to

work with streaming or batch data. Data ingestion can happen from applications and gateways into an event hub or IoT hub. The event or IoT hub then streams the data into Stream Analytics for real-time analysis.

Batch systems process groups of data stored in an Azure Blob store. They do this in a single job that runs at a predefined interval. You should not use batch systems for business intelligence systems that can't tolerate the predefined interval, for example, an autonomous vehicle can't wait for a batch system to adjust its driving. Similarly, a fraud-detection system must decline a questionable financial transaction in real time.

Azure Synapse Analytics

Synapse Analytics (formerly Azure SQL Data Warehouse) is another type of analytics service that provides a unified experience for big data and data warehousing. It mixes the data warehousing and big data as a single unit of service. Doing this gives organizations the ability to analyze and query large volumes of structured and unstructured data using SQL, Spark, and Power BI. Using Synapse SQL, you can make T-SQL queries to distribute query systems. This allows the option to perform data warehousing use cases for data visualization. Furthermore, the support for T-SQL help addresses the live analytics streaming and machine learning scenarios.

Using Synapse Analytics, you can integrate these two and query data on your terms, using serverless or dedicated options. Azure Synapse combines these worlds with a unified experience to ingest, explore, prepare, transform, manage, and serve data for immediate BI and ML needs.

As noted, Azure Synapse Analytics is the next generation of Azure SQL Data Warehouse. It enables you to transform and load from any data source, both relational and non-relational databases. Whether you are doing this on premises or in Azure, it unifies all the data and lets you process and analyze it using the SQL language. **Azure Synapse Studio** is an excellent tool for creating a workspace to ingest your big data, develop with data flows, create PowerBI reports, setup data integration pipelines, and more.

Azure Databricks

Databricks is a quick, manageable, and combined Apache Spark-based analytics platform. It delivers a collaborative workspace for data engineers, data scientists, and data analysts. With Azure Databricks, organizations can efficiently process and analyze big data using Apache Spark, Python, and R, and build and deploy ML models at scale. Using this service also lets you set up managed Apache Spark clusters with autoscaling and autotermination, eradicating the complexity of setting up Spark in your regional data center.

Azure Data Lake Storage

Azure Data Lake Gen2 is designed for enterprise big data analytics with dedicated features and capabilities. It is built on Azure Blob Storage and includes the features and elements of [Azure Data Lake Storage Gen1](#).

Data Lake Storage Gen2 provides features to enhance security, performance, and management. It supports file system semantics, file-level protection, and scaling options. Since it is built on top of Azure Blob Storage, there are flexible options for storage types in different tiers. There are also low-cost options and support for high availability and disaster recovery.

Azure HDInsight for Hadoop, R Server, HBase, Spark, and Storm Clusters

The free, open source computing platform known as [Apache Hadoop](#) has been a major player in the field of big data, and its ecosystem is still growing and developing, despite the fact that Hadoop itself is not very well known. It also offers several compelling application cases and allows you to carry out complicated, distributed analysis operations on practically any volume of data.

Azure HDInsight is an open source, managed analytics and cloud-based service for users who require more significant and broader support for

analytics capabilities for big data, for example, organizations that need to process vast quantities of data with streaming or historical data capabilities.

Using HDInsight, you can create significant data clusters using Hadoop and scale them based on your demands and workloads. It allows you to implement Hadoop analytics for existing data by integrating with other services and tools like Azure Data Lake Storage and Azure Data Factory.

The support for Hadoop benefits those who need to use the capabilities of Hadoop tooling like HBase, Apache Spark, Storm, Hive, and Apache Kafka to process and analyze big datasets.

In addition, Azure HDInsight provides monitoring, high availability, security, and compliance at an enterprise scale in Azure. You will benefit from Azure HDInsight if you utilize custom code with these frameworks.

HDInsight also gives you control over your cluster setup and the software installed on them. HDInsight is a recommended alternative if you migrate Hortonworks, Cloudera, or MapR collections from on-premises environments or other clouds.

Overall, Azure HDInsight can be used for a variety of big data scenarios, including historical or real-time data used for data warehouses, data science, IoT, and hybrid data solutions.

Azure Data Factory

Azure Data Factory is a data integration service on Azure that enables developers to create, schedule, and manage data pipelines capable of moving and transforming data from various sources and destinations. Developers can perform data integration duties such as ETL from on-premises and cloud-based data sources, including SQL Server, Oracle, MongoDB, Salesforce, Azure Blob Storage, and others. The service is compatible with **Hadoop Distributed File System (HDFS)**, Apache Spark, and Microsoft Azure HDInsight.

Developers can construct data pipelines using Data Factory's user interface or code-based authoring tools such as JSON, Python, and .NET. Once the

data pipelines have been constructed, they can be scheduled and monitored to ensure the data integration processes operate efficiently and reliably.

Data Factory offers a variety of integration options with other Azure services, such as Azure Functions, Azure Logic Apps, Azure Stream Analytics, Azure Databricks, and more, to enable the creation of complex data workflows capable of handling real-time data streaming and data orchestration.

Azure Analysis Services

Analysis Services is an enterprise-grade data model in the cloud. It is ideal for an enterprise that needs an analytics engine as a managed cloud service. It can be **configured using infrastructure as code tools** like the Azure Resource Manager (ARM) template, Azure Bicep, and Terraform. You have the option to combine data from multiple sources and build them as one semantic model.

Analysis Services allows you to develop high-performance BI solutions with secure access and fast time to delivery. It can be configured to perform autoscaling based on the analytical workload, and you pay only for the resources you consume. Analysis Services also lets you import existing tabular models.

Figure 7-3 shows the different kinds of solutions and tools used for analysis in Azure.

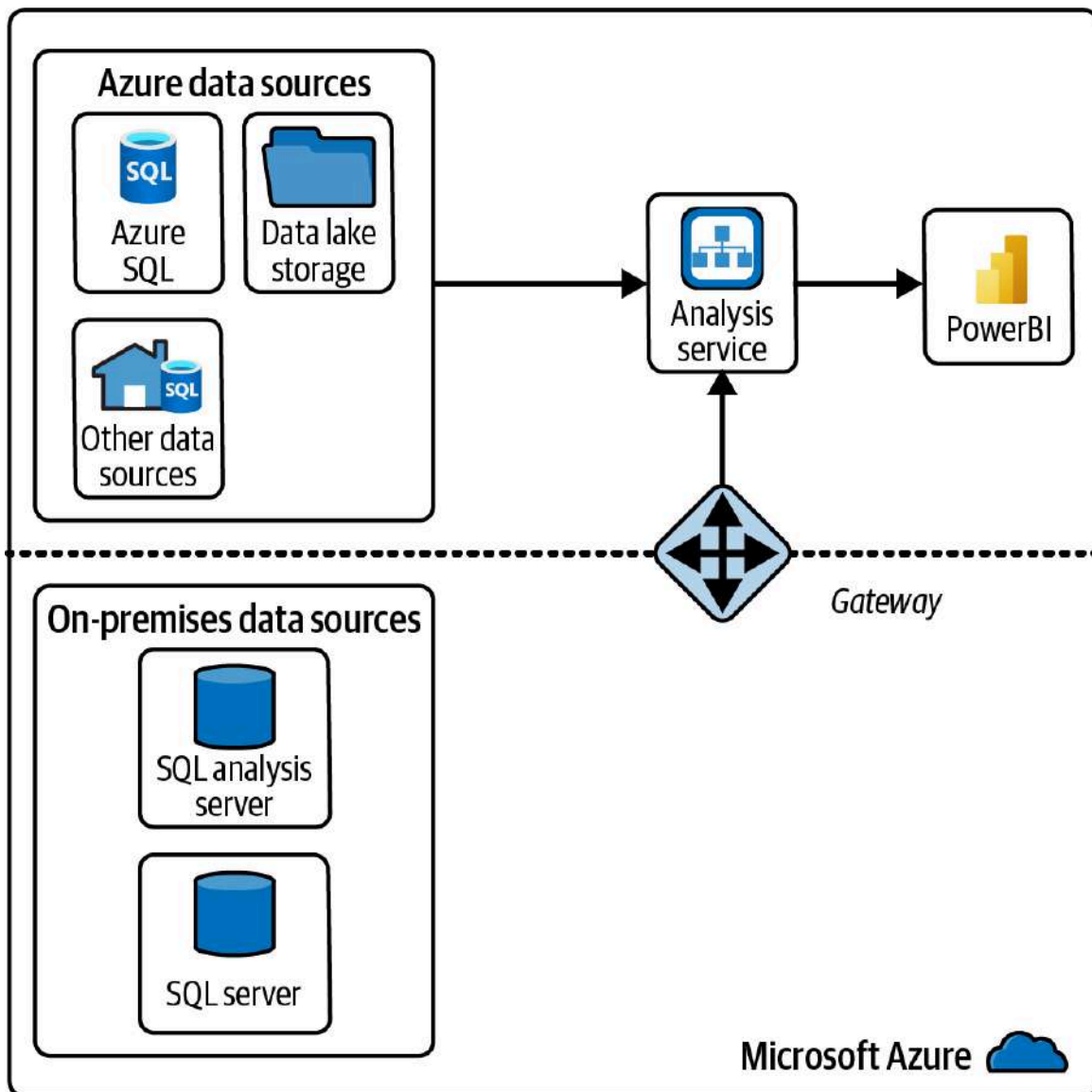


Figure 7-3. Azure Analysis Services

Power BI Embedded Analytics

Microsoft Power BI is a leading data analytics service in Azure. Many people are familiar with Power BI because they interact with reports in the Power BI service or Power BI for mobile apps. Those working with data analytics and reporting can use Power BI efficiently.

In addition, cloud engineers and software developers can use any Power BI content embedded in their applications by programming it. Developers can programmatically embed Power BI analytics in applications to visualize and

present data or reports. The end users of the Power BI embedded analytics can easily view the actual data and reports and make decisions based on facts and statistics, as shown in **Figure 7-4**, which illustrates Power BI Embedded Analytics.

You can embed Power BI content in any app by using an HTML iframe element. It is purpose-built for developers, and it can be used with client REST APIs; there are SDKs for different programming languages and platforms, such as .NET, JavaScript, and TypeScript.

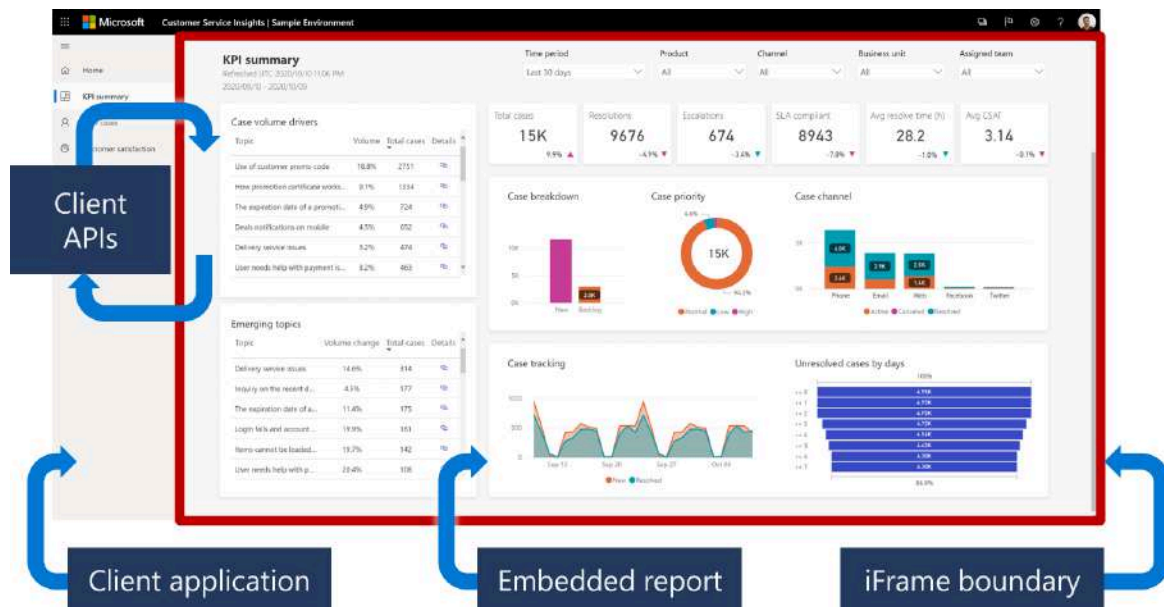


Figure 7-4. PowerBI embedded analytics reporting dashboard. Image credit: *Microsoft*

If you are a developer who wants to learn more about how you can enrich your data analytics and reporting using Power BI embedded analytics, check out the Microsoft article **“Embed a report in a secure portal or website”**.

Microsoft Purview for Data Governance

Microsoft Purview (previously Azure Purview) is a solution in the governance portal that provides a suitable data governance service to help users manage their on-premises, multi-cloud, and SaaS data.

Purview helps automate data discovery through data classification and scanning as a service for assets across your data estate. The metadata and descriptions of the discovered data assets can then be integrated into your data estate's map. Purpose-built apps create environments for data discovery, access management, and insights.

Microsoft has documented **Microsoft Purview account architectures and best practices** to help with centralizing your data governance solutions. Additionally, the **Microsoft Purview governance portal** provides features for governing data, such as the Data Catalog, Data Estate Insights, Data Sharing, and discovery challenges for data consumers.

TIP

If you want to try the different solutions and features of Microsoft Purview, a 90-day Microsoft Purview solutions trial is available. It is a good option for your organization to explore its capabilities for compliance needs. Microsoft 365 E3 and Office 365 E3 customers can start now at the Microsoft Purview **compliance portal trials hub**.

Microsoft Purview Data Lifecycle Management provides the features and tools necessary to retain the content you need to keep and erase the content you don't. Frequently, regulations and compliance standards demand that content be managed and deleted. However, eradicating content lacking business value may assist in risk and liability management by reducing compliance and security risks.

Final Note on Data Management and Analytics in Azure

Organizations that desire quality analytics should work with accurate, trustworthy, and robust data. Data and information security practices are highly recommended to ensure your enterprise data is **secured both in transit and at rest**.

Taking advantage of **Azure Data Encryption at rest** and using different types of data encryption models are a few of the tools offered by Microsoft Azure to protect and secure your enterprise data according to your organization's data security and compliance requirements.

Learn By Doing (Try It!)

In addition to the supplementary hands-on **lab repository** for the topics in this chapter, the following tutorials are recommended as they are updated based on Microsoft's technical updates for the service:

- **Azure Synapse Analytics for data engineers**
- **Quickstart: Share and receive Azure Storage data in-place with Microsoft Purview Data Sharing**
- **Quickstart: Create an account in the Microsoft Purview governance portal**
- **Model, query, and explore data in Azure Synapse**
- **Create Apache Hadoop clusters using the Azure REST API**

Summary

In this chapter, you learned about the fundamental concepts of big data, including its characteristics, common uses and benefits, and its importance to data science, analytics, ML, and reporting. You also learned about Azure's different big data, analytics, and reporting services.

Finally, you learned about Microsoft Purview's solutions in the governance portal, a data governance service that helps you manage your on-premises, multi-cloud, and SaaS data.

Check Your Knowledge

1. Big data, data science, and analytics are evolving and trending. Why do you think they are essential?
2. Big data and analytics tools without good cybersecurity protection could put an organization at security risk and leave it vulnerable to data breaches or cyberattacks. What can an organization do to prevent this from happening?
3. What are the common challenges in building big data solutions?
4. Apache Hadoop is used in big data analytics solutions. Why do you think Hadoop is commonly used in big data analytics?
5. How does Azure Synapse Analytics differ from Azure Data Lake Analytics?

For the answers to these questions, see the [Appendix](#).

Recommended Learning Resources

“Azure Analysis Services Documentation.” Microsoft Learn, <https://oreil.ly/06-q9>.

Basak, Anindita, Krishna Venkataraman, Ryan Murphy, et al. *Stream Analytics with Microsoft Azure*. Birmingham, UK: Packt Publishing, 2017.

“Big Data Architecture Style.” Microsoft Learn, <https://oreil.ly/z1MwG>.

“Cloud-Scale Analytics.” Microsoft Azure, <https://oreil.ly/hlGO6>.

Cote, Christian, Michelle Gutzait, and Giuseppe Ciaburro. *Hands-On Data Warehousing with Azure Data Factory*. Sebastopol, CA: O’Reilly Media, 2018.

“Data Science with Microsoft Azure.” Pluralsight, <https://oreil.ly/-Z3Fu>.

Greenaway, Jasmine, Dmitry Soshnikov, Nitya Narasimhan, et al. “Data Science for Beginners – A Curriculum.” GitHub, <https://oreil.ly/8MbS2>.

“Introduction to Data Analytics on Azure.” Microsoft Learn,
https://oreil.ly/_uQoU.

Microsoft Mechanics. “An Introduction to Azure Analysis Services.”
YouTube video, March 20, 2017, <https://oreil.ly/lzFGL>.

“Microsoft Purview.” Microsoft Learn, <https://oreil.ly/-sbUo>.

“Modern Analytics Architecture with Azure Databricks.” Microsoft Learn,
<https://oreil.ly/W-JFR>.

Mount, George. *Advances into Analytics*. Sebastopol, CA: O’Reilly Media,
2021.

“Overview: Big Data Cluster Security.” Microsoft Learn, December 2,
2019, <https://oreil.ly/Gftqv>.

¹ Kalev Leetaru, “How Do We Define *Big Data* and Just What Counts as a *Big Data* Analysis?”
Forbes.com, January 9, 2019, <https://www.forbes.com/sites/kalevleetaru/2019/01/09/how-do-we-define-big-data-and-just-what-counts-as-a-big-data-analysis>

Chapter 8. Cloud IoT and Maps Services

The Fourth Industrial Revolution, or Industry 4.0, introduced smart automation and increased interconnectivity in the industry. IoT technologies are a great enabler for Industry 4.0, and Azure cloud allows you to build secure, compliant, and scalable IoT solutions.

— **Goran Vuksic**, CTO and Co-founder of
SyntheticAIData, Microsoft MVP for AI

In **Chapter 7**, you learned about the different solutions and services to plan, design, and build solutions for big data, analytics, and reporting in Azure. Your knowledge of analytics and reporting will apply to handling metrics and telemetry data gathered from IoT devices for the cloud.

In this chapter you will learn about the IoT, maps, and edge services in Azure. After reading this chapter, you will understand the Azure services you can use in different use cases to develop IoT solutions.

Internet of Things

The [Wikipedia.org](https://en.wikipedia.org) defines the IoT as:

Internet of Things (IoT) extends Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet. They can be remotely monitored and controlled.

Today's architectures feature highly scalable event handling, which enables real-time analysis in what Microsoft has named the intelligent cloud. They support the deployment of machine learning at the intelligent edge in the devices.

As more advanced IoT solution components and capabilities become available, previous architecture patterns have evolved to take advantage of these new capabilities and enable the deployment of more sophisticated business solutions.

IoT devices are growing in popularity and are evolving. Smart lamps, smart wearables, and temperature sensors are becoming essential to our daily lives and routines. Smartwatches help us get moving and exercise more efficiently by giving us data and statistics on how we perform. Smart homes and lamps help us control our home and office lighting and appliances without manual work. Temperature sensors allow us to monitor the temperature of our home and office regardless of location.

An IoT Analytics report from Spring 2022 indicates that the number of IoT devices is projected to grow and devices are expected to be connected globally.

Aside from the practical uses of IoT devices for humans, they also help societies around the globe through smart retail and smart city solutions. An IoT system consists of interconnected computing devices: mechanical and digital devices, objects, animals, or people that are provided with individual identification numbers and the capability to send and receive data over the internet. Examples of this system include the Internet of Vehicles (IoV), the Internet of Buildings (IoB), and the Internet of Agriculture (IoA).

The IoT makes virtually everything smart in that devices are able to do tasks on schedule or based on the conditions we set, and they learn. It helps improve the power of data collection, AI algorithms, networks, human interaction, etc. IoT technologies can also help monitor the health of a person with diabetes or assist pet owners by using IoT GPS-tracking sensors to locate lost pets.

The devices that implement IoT operations collect data telemetry that can be stored and hosted in Azure storage services provided by Microsoft Azure. Big data, reporting, and analytics tools are also helpful in visualizing the collected data.

Making Sense of IoT Technology

The IoT helps us connect and exchange data with the sensors of these objects. It promises and envisions a digitalized future where devices can communicate to assist and enhance our daily routines.

Figure 8-1 shows us the different components of the IoT, which is essential to understand as an overview of its workflow and data processing before it gets to the user.

These IoT devices have sensors embedded in them, giving them the capability of sensing their environment and collecting data. For example, IoT devices with sensors check the current temperature and humidity indoors or outdoors.

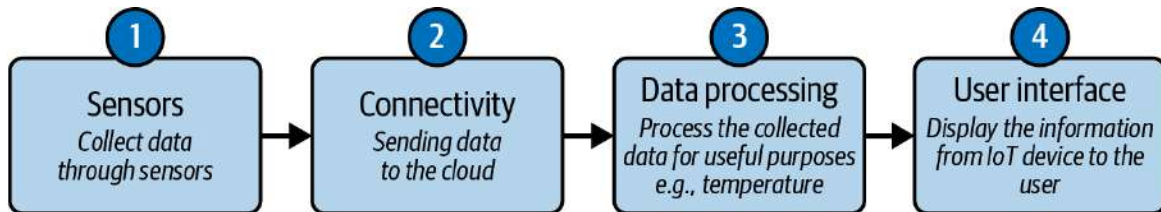


Figure 8-1. Components of IoT: sensors, connectivity, data processing, and user interface

IoT devices, such as applications, smoke detectors, mobile phones, robot vacuum cleaners, etc., store this information from their sensors in the form of telemetry data with different properties, units, values, etc.

The sensors embedded in IoT devices are connected to an IoT platform and continuously transmit data for further processing. The IoT platform then gathers all the valuable data collected by these IoT devices, which is processed/transformed into helpful information. The data can then be visualized through an application or a user dashboard.

To manage and collect the data from these smart devices, the IoT platform includes a management interface for its users, cloud servers, databases, storage, etc. It manages the collected data by integrating and processing the information.

Furthermore, the platform analyzes the data thoroughly to gather essential details and then sends instructions based on the data provided. This data

cycle is not limited to one IoT device transmitting data back and forth to the IoT platform. The data is aggregated and shared with other IoT devices to improve performance and user experience. As a result, we need to consider the privacy issues of using these IoT devices.

We will address the security concerns and challenges in IoT technologies later in this chapter.

Components of an IoT Solution

Table 8-1 shows the different IoT components used in developing an IoT solution.

Table 8-1. Components of IoT

Component	Description
Sensors & IoT devices	Smart devices with IoT sensors help us gather valuable data, which can vary in type and complexity. There are different sensors for temperature monitoring, smoke detection, motion detection, suspect behavior sensor on security cameras, etc.
Connectivity	Connectivity means the data collected is synced from the IoT device sensors to the cloud for further processing. The typical communication transmission mediums include satellite networks, mobile networks, Bluetooth, Wi-Fi, WAN, etc.
Data processing	When the data is collected, it goes to the cloud platform for data processing. However, data processing from IoT devices can be very complex, like detecting and trying to identify objects using computer vision on a video surveillance camera.
User interface	For visualization of collected and transformed data, an application or user interface dashboard is needed for further analysis, reporting, or perhaps processing for event notification based on certain conditions or rules.

These applications can be developed to be event-driven, which means that they can be configured so that the user receives notices, critical alarms, etc., when a significant event happens or a parameter is exceeded. For example, an intelligent video security camera can send an SMS, email, or trigger a call with an alarm if an intruder and suspicion of robbery are detected while

the user is away from home. The user can also perform some interaction with the IoT device depending on the type of application and its complexity.

The components of an IoT solution are more comprehensive than just sensors and devices. The IoT solution also requires connectivity, data processing, and a user interface, which is easier to implement in a cloud-based solution.

Different Types of IoT Applications

Table 8-2 lists useful applications of IoT solutions; most of these are hosted in the cloud. The list of examples will regularly change.

Table 8-2. IoT Solutions Applications

Application	Description
Smart thermostat systems	Helps save on power usage bills by knowing your usage patterns for heating during the winter and air conditioners during the summer.
Activity trackers	Capture our heart rate patterns, calorie expenditures, activity levels, skin temperatures, etc.
Connected cars	Handle bill management systems, create features that allow drivers to access parking, update maps, manage driver profiles, and improve safety features.
Smart electronic plugs	Track a device's energy level and provide custom notifications and useful electricity analytics to your smartphone
Smart home	Automate the schedules of home IoT devices, for example, schedule smart lights based on sunset or sunrise or get notified in real time of a possible emergency, such as a smoke detector alarm.

Based on the number of IoT applications and their many applications, we expect this technology will continue to evolve along with cloud computing development with edge computing, AI, and ML.

Challenges of IoT

IoT technology, its benefits, and its promising use cases generate opportunities for many enterprises, industries, and organizations to use

devices in a smarter way, automate them, and use IoT telemetry data to create tools and solutions.

The advantages of IoT

Here are the common advantages of IoT solutions:

Technology innovation

IoT technology enables us to learn more about devices and how we can further develop them. Using IoT, for example, a manufacturer can use car sensors to collect data to analyze and improve the design and features of a car to make it safer for drivers and passengers.

Customer satisfaction and engagement

IoT technology improves customer satisfaction by detecting problems and improving the processes, speed, and efficiency of communication supporting customers, which leads to customers using the technology more.

Reduced waste and resource sustainability

IoT's real-time information leads to effective decision making and management of waste at the industry level, helping to achieve sustainability goals. For example, if a manufacturer sees an issue in several car engines, it can track those engines' manufacturing plans and solve this issue with the manufacturing belt. Furthermore, waste management using IoT helps improve the efficiency of waste and recycle management, such as having sensors that help reduce the overflows of waste, scheduling management of waste truck pickups, sorting different waste materials, and many other examples

Real-time data collection and streams

Traditional data collecting is intended for inactive use: data is stored and users must retrieve it to process and analyze. IoT technologies allow quick action actively and in real time.

IoT in industrial farms

Smart farming, or digital agriculture, is an excellent example of how IoT works on an industrial scale. Smart farms have devices that provide real-time data about crop conditions, such as moisture balance, soil texture, pesticide levels, and more.

Employee monitoring systems

IoT is incorporated into the business sector through employee monitoring systems such as facial recognition trackers, fingerprint scanners, and iris scanners. Some companies install these devices at entrances to their premises to track employee attendance and provide security.

IoT warehouse

With the increased demand for consumer goods, using IoT in inventory management and control reduces manual tracking time and saves money.

The disadvantages of IoT

The following are some of the disadvantages and essential things to be mindful of, whether you are building IoT solutions in your organization or using them as a consumer.

Security

IoT systems can be prone to security risks if the IoT platform and devices are not configured with sufficient security authentication, network, and endpoint protection.

Privacy

The use of IoT exposes a substantial amount of personal data, in detail, without the user's active participation. The fact that some personal data is being collected along with other data creates many privacy issues.

Flexibility

Because diverse systems are involved, there is considerable concern regarding the flexibility of and integrations among IoT systems.

Complexity

IoT system design is complicated. To design and build an IoT system, you need to identify the business needs, create a plan, investigate what kind of IoT devices or types of sensors you need, and learn how to configure them with your business operations. Moreover, you need reliable IoT management platforms or tools for effective deployment and maintenance.

Legality and compliance

Legality and compliance are one of the interesting subjects for discussion. Compliance is a recurring challenge for organizations using IoT solutions. The two primary issues that make compliance with IoT challenging are the lack of visibility into what's happening with the IoT devices deployed in the field in different industries as well as the absence of well-defined IoT compliance standards being developed and shared to the public for education and information.

In the 2020 white paper “**Why IoT projects fail**”, Beecham Research estimated that nearly three-quarters of IoT projects wouldn’t be successful. Top reasons included that businesses needed better IoT solution strategy planning, lack of technical expertise, and unforeseen technological problems.

One known challenge is the inability to use IoT telemetry data and business processes effectively. Another challenge is establishing technology standards and dealing with data privacy, compliance, and security risks.

TIP

To learn more about the general cybersecurity standards of IoT solutions, see the **IoT Security Compliance Framework**, which will guide you on making your IoT solutions secure and compliant.

Microsoft’s Well-Architected Framework is suitable for learning more about the best security and compliance practices in IoT.

To tackle these challenges, it is crucial to have a solid organizational strategy for implementing IoT solutions.

IoT in Microsoft Azure

Microsoft Azure’s IoT platform can pre-customize or fully customize solutions based on business requirements. The Azure IoT platform provides options for companies both early in their IoT strategy development and when their implementation is more advanced.

Cloud platforms can help quickly scale IoT systems to include IoT-supported devices from different manufacturers. They also provide support for advanced data analytics and ML capabilities.

Azure IoT

Azure IoT is a collection of managed service platforms across cloud and edge computing. It helps users to connect, monitor, and manage billions of

IoT devices from different locations.

The Azure IoT platform is built with security, performance, data management, and analytics that will aid organizations, businesses, and their users in deploying and managing their IoT applications for end users.

Azure IoT has three components:

Things

Things are physical devices or objects connected to the cloud persistently or intermittently. Examples are industrial equipment or appliances with sensors.

Insights

Insights are information collected by the physical IoT devices or things. They are processed, analyzed, and turned into actionable knowledge by people working with them or using AI.

Actions

Actions are when users do something about the insights provided through the IoT platform and then applies them to their business operations, existing applications, or tools.

These three components are essential in an Azure IoT service because they can be designed and built using different cloud technologies based on your use case.

For more information, see the open public website [AppSource](#), which publishes a list of applications built by IoT solutions.

Azure IoT Hub and its device provisioning service

The IoT Hub is a central hub where you can manage and control the communication between your IoT devices and their applications. Nearly any device can be embedded with sensors, and software to capture data can

be added via an internet connection. If you are working with IoT development in Azure, IoT Hub and its IoT device provisioning service might be something worth considering.

When you connect any IoT device in the cloud as an IoT service, you also collect helpful telemetry and data, like temperature, humidity, etc. This data telemetry will be sent in real time to a central place where you can manage the data and use it to create solutions, for example, integrating it with event-driven and serverless solutions within Azure.

Azure IoT Central

Azure IoT Central is a secure IoT application platform that scales as your business demand increases. This Azure service for IoT solutions helps organizations by removing the hassle of creating IoT solutions from scratch. This platform also helps manage IoT at the enterprise level in-house.

If you need to connect and manage millions of IoT devices across the globe, using Azure IoT Central is ideal. It enables you to provision your devices, visualize the data telemetry collected, and monitor them with the capability of integrating other valuable services for alerts and communication.

Developers can use the **IoT Central REST API** to develop new applications or integrate existing ones. Using the REST API for IoT application development, you can provision your devices and manage their operations.

Azure IoT Plug and Play app

Using Azure IoT services, you can connect virtually to any IoT device or a simulator using your mobile phone. IoT Plug and Play is an app for Android and iPhone that can turn your phone into an IoT device. This is especially handy if you want to get started with an IoT device but are still waiting to invest a lot of money. Using IoT Plug and Play, you can quickly explore IoT features without configuring a dedicated device. This is ideal if you are learning IoT development in the cloud.

The app also takes telemetry data from your smartphone sensors and allows you to send commands to it from the cloud-based IoT service on Azure.

Check out this [guide](#) from Microsoft to get started using your smartphone as an IoT device.

Azure IoT DevKit and Azure-accredited IoT devices

The MXCHIP DevKit is an all-in-one IoT development kit designed and optimized for users or developers who want to create prototypes and develop IoT solutions leveraging services in Azure and other third-party services.

Figure 8-2 shows the different sensors included in the kit. Essentially, it's a smart IoT device compatible with Arduino, an open source prototyping platform for building digital devices.

Its user interface board includes features such as an OLED display for data, different sensors, a hardware debugging chip (ST-Link), and a security chip. There are [sample IoT projects](#) available with documentation that will help you get started with this device.

This IoT DevKit can also use IDE coding tools like Visual Studio Code with Arduino Extension to quickly build a full-fledged IoT application. The kit integrates with multiple services like IoT Hub, Logic Apps, and Cognitive Services.

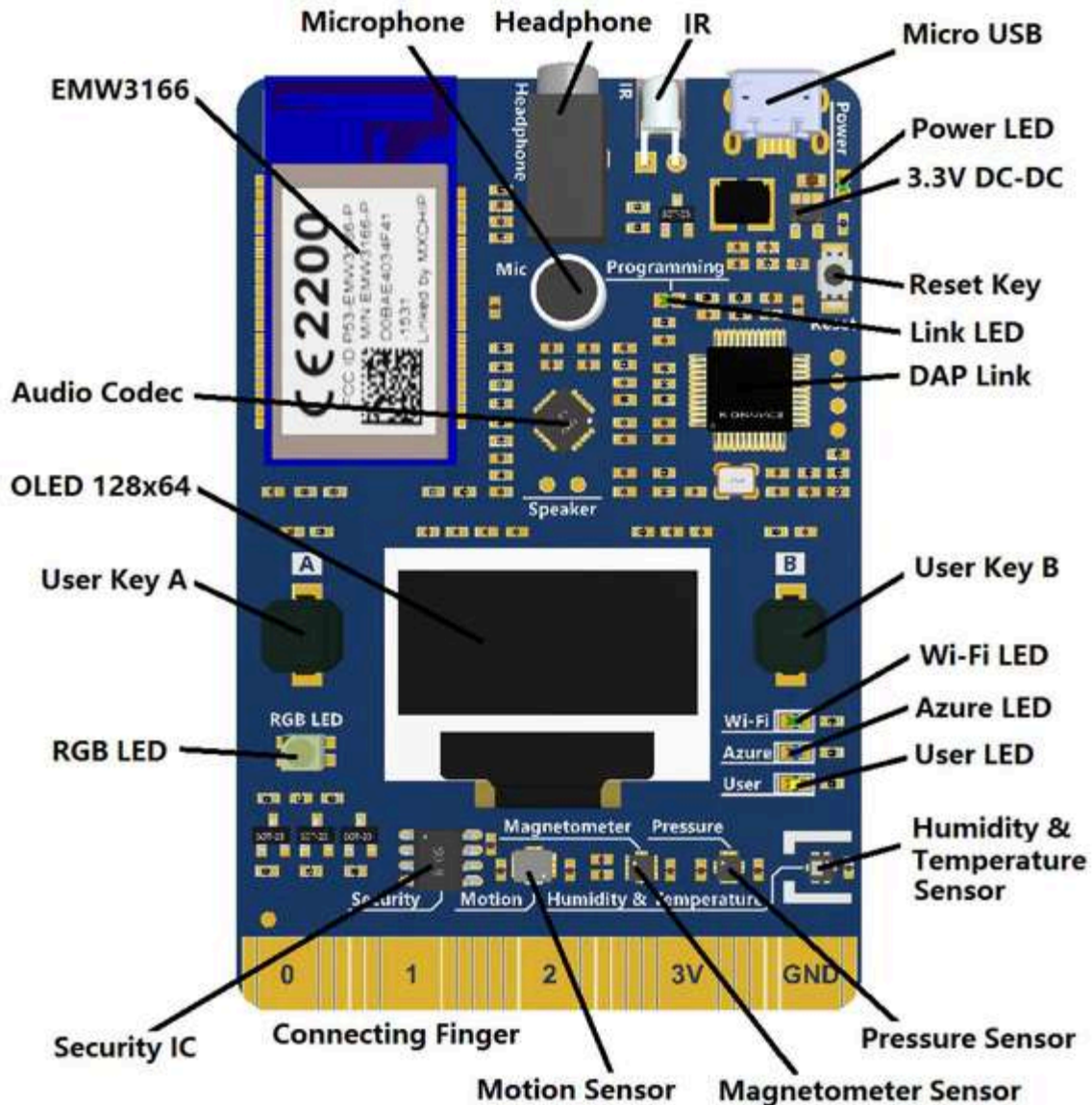


Figure 8-2. IoT DevKit sensors (image credit: *Azure IoT DevKit official documentation*)

Digital Twins

Digital Twins is an IoT platform that will help you create and design a digital view of real-world things such as places, people, business processes, etc. This platform for IoT also helps you gain insights to help you deliver better products, optimize operations and costs, and create breakthrough customer experiences.

Digital Twins is a platform as a service (PaaS) that will help you develop twin graphs based on digital models of entire territories, including

buildings, industrial factories, farms, energy networks, stadiums, railways, etc. Since the models created are digital, you can gain insights that will optimize business operations, reduce costs, and provide satisfactory customer experiences.

You can also develop a digital twin that reveals IoT devices in the broader cloud solution. This can be connected to IoT Hub device twins to send and receive live data and telemetry.

NOTE

Although it is easy to confuse them, the IoT Hub device twins differ from Azure Digital Twins. Although the IoT Hub supports IoT Hub device twins for each IoT device you connect to, digital twins in Azure can be anything defined by digital models and instantiated within Digital Twins.

Usually, IoT device twins are composed of JSON documents that typically contain information about the state of a device, such as device configuration and metadata.

Example 8-1 is a code snippet that illustrates how a device twin is represented in JSON format. The JSON file is composed of properties like the ID of the device twin, serial number, and other helpful metadata.

Example 8-1. Device twin in JSON format

```
{
  "deviceId": "example-device-twin",
  "modelId": "dtmi:com:example:TemperatureController;1",
  "version": 15,
  "properties": {
    "desired": {
      "thermostat1": {
        "__t": "c",
        "targetTemperature": 22.8
      },
      "$metadata": {...},
      "$version": 4
    },
    "reported": {
      "serialNumber": "deviceserialnumber77899",
      "thermostat1": {
        "maxTempSinceLastReboot": 25.3,
```

```

    "__t": "c",
    "targetTemperature": {
      "value": 21.8,
      "ac": 200,
      "ad": "Successfully executed patch",
    }
  },
  "$metadata": {...},
  "$version": 11
}
}
}

```

Example 8-2 is a code snippet that illustrates how a digital twin is represented in JSON format. Notice that the JSON file is composed of properties almost like the **Example 8-1** except the property `serialNumber` is defined differently.

Example 8-2. A code example for Azure Digital Twins in JSON format

```

{
  "$dtId": "example-digital-twin",
  "serialNumber": "deviceserialnumber77899",
  "thermostat1": {
    "maxTempSinceLastReboot": 25.3,
    "targetTemperature": 21.8,
    "$metadata": {
      "targetTemperature": {
        "desiredValue": 21.8,
        "desiredVersion": 4,
        "ackVersion": 4,
        "ackCode": 200,
        "ackDescription": "Successfully executed patch",
        "lastUpdateTime": "20220-11-17T06:11:04.9309159Z"
      },
      "maxTempSinceLastReboot": {
        "lastUpdateTime": "2022-11-17T06:10:31.9609233Z"
      }
    }
  },
  "$metadata": {
    "$model": "dtmi:com:example:TemperatureController;1",
    "serialNumber": {
      "lastUpdateTime": "2022-11-17T06:10:31.9609233Z"
    }
  }
}

```

For additional examples and to learn more, see the [documentation on managing IoT Plug and Play digital twins](#) and the [Digital Twins API](#).

When implementing and using Digital Twins, you're typically charged based on consumption, and pricing is based on query units, operations, and messages.

An IoT developer tool, Azure Digital Twins Explorer, can interact with and visualize your Azure Digital Twins instance data, such as twin graphs.

Securing IoT on Azure using Defender for IoT

In an earlier section of this chapter, we discussed security and compliance considerations as a few of the challenges of IoT technology. If your IoT solutions are hosted on Azure, [Defender for IoT](#) helps secure your IoT solutions.

Defender for IoT helps organizations automatically discover and detect unsecured and unmanaged assets in their IoT devices or solutions. Additionally, the built-in security features, such as security posture management, endpoint threat detection, and IoT Hub integration, help protect your IoT services and solutions in the cloud.

Defender for IoT helps by detecting critical vulnerabilities, security threats, and possible real-time anomalies. It provides quick, informative insights leveraging analytics and ML for suspicious behaviors. These insights help minimize the need for manual setup.

Furthermore, Defender for IoT also includes network sensors on premises to capture the traffic from or to devices. The sensors are then connected to a port or a tap, and they can begin to monitor network communications for any anomalies.¹ Network analyzers play a crucial role in protecting IoT devices and their associated networks against security threats and attacks.

These analyzers monitor and examine the traffic on a network for signals of malicious behavior, such as infection by malware or attempted network intrusions. As a result of conducting this network monitoring, businesses

can determine the origin of the assault, the nature of the attack, and the particular IoT devices that have been compromised.

Additionally, network analyzers can be used to assess the execution of IoT networks, such as the reliability and velocity of data transmission, and identify any bottlenecks or areas for improvement. They help optimize the performance of IoT networks and make them more secure and resilient.

Another use case for network analyzers in IoT security is to monitor and analyze the wireless signals emitted by IoT devices, such as Wi-Fi or Bluetooth signals, for signs of unauthorized access or interference. Typical IoT attacks can be avoided using these analyzers.

Overall, network analyzers are essential tools in defending IoT networks against security threats and attacks, providing critical information and insights that help improve the security and performance.

Azure Maps

Azure Maps provides a collection of cloud-based mapping services that help with geospatial assistance. The software development kits, or SDKs, of Azure Maps are used to map data. This is helpful when creating geographic features for web or mobile applications. These services include components that help manage data types like route, geospatial, map search, and rendering.

Features of Azure Maps include:

Geospatial services

A set of REST APIs that allows developers to perform geospatial operations such as geocoding, reverse geocoding, routing, and traffic data.

Interactive maps

A web-based map control that allows developers to quickly add interactive maps to their web applications.

Location APIs

A set of REST APIs that provides access to data related to points of interest, geofencing, and other location-based information.

Spatial operations

A set of REST APIs that allows developers to perform complex spatial operations such as geofencing, buffering, and spatial analysis.

Real-time traffic

A set of REST APIs that provides real-time traffic information, including incidents, congestion, and traffic flow.

Mobility services

A set of APIs that provides information related to mobility, such as public transit information and electric vehicle charging locations.

If you want to develop solutions with Azure Maps, a free account for this service is available to help you get started. It also has web and mobile development SDKs, which allow developers to add map services to their new or existing applications.

WARNING

Azure Maps is available in most countries and regions except parts of China and South Korea. When you are hosting or deploying your applications for Azure Maps, ensure that you choosing an Azure region that is currently supported.

The **Maps Power BI visual** supports the visualization of spatial map data, for visualizing business data based on the location context or categories. Related to this topic, it is worth exploring **hyperconverged analytics** where the methods for traditional business intelligence (BI) can be empowered by AI models using real-time data.²

In addition, Azure Maps can also be implemented and developed with low-code/no-code tools, for example, in Power Apps. You can create an interactive map with **canvas apps**.

Control Results of Azure Maps with Geographic Scope

Azure Maps has localization support, a map service that is supported globally; this allows you to specify a **geographic scope**. By determining your geographical area, you can limit data residency due to compliance regulations. For example, you can select the data residency for the European (EU) or the United States (US) geographic areas.

All requests are typically stored in the specified geographic area. A cloud provider like Microsoft also supports replicating customer data to other Azure regions within the exact geographic location to fulfill high availability and disaster recovery.

For example, if you are using Azure Maps European API as your geographical endpoint, your map requests or data resides in a European Azure data center. This API allows end users or consumers to choose where their map data can be accessed.

Azure Maps European API provides high-quality, reliable, and accurate data for mapping and location-based services in Europe. You can integrate it into different applications, which are highly scalable and protected, making it a recommended choice for organizations looking to build mapping-based applications in Europe.

If you are in Azure geographic areas like Europe, your API geographical endpoint would be *eu.atlas.microsoft.com*. If you are in the United States region, your ideal API geographical endpoint would be

us.atlas.microsoft.com. When using the **Azure Government cloud**, use the *atlas.azure.us* endpoint.³

Azure Maps provides a set of REST APIs for various geospatial operations, including geocoding, routing, and traffic data. These APIs are available globally and can be used to build applications in any country.

The API provides data for mapping and location-based services for countries worldwide. The scope of the Maps API varies by country, so it is critical to review the detailed and updated Microsoft Learn **documentation** for the country you are interested in for more information about the general data and services.

Authentication and Security on Azure Maps

Azure Maps provides several mechanisms for authentication and security:

Authentication via Microsoft Entra ID

Microsoft Entra ID is the default authentication mechanism for Azure services. The ID authentication support in Maps allows you to authenticate user credentials.

API key authentication

Authentication using API keys is the simplest way to authenticate requests to the Azure Maps API. You can generate API keys in the Azure Portal and then pass the key as a query parameter in your API requests.

OAuth 2.0 authentication

Since Maps supports OAuth 2.0 authentication, you can gain secure access to the API on behalf of your users. This is helpful if you intend to utilize the API to gain access to data that is restricted to a certain user or if you want to use Azure Maps as part of a larger application.

In addition to these authentication options, Azure Maps also has the following security features to safeguard data and applications in Azure:

Secured encryption via TLS/SSL encryption

Using this security option, all API requests and responses are encrypted using TLS/SSL to protect sensitive information transiting the internet.

Network security groups (NSGs)

NSGs are used widely in networking in Azure. Implementing NSGs allows you to control the inbound and outbound traffic to and from your Azure Maps account. You can define rules that allow or block the traffic based on destination source, IP addresses, ports, and protocols.

Role-based access control (RBAC)

RBAC allows you to control access to Azure Maps by defining roles and permissions for users. It is possible to assign roles to users and groups and then specify what actions they can perform with the API.

Azure Maps provides potent security elements to protect your applications and data. This cloud service supports high scalability and reliability, essential in designing and developing applications with map integration.

Maps Integrations with Azure Event Grid

The Azure Maps service integrates well with Event Grid if you want to handle events and notifications. It is a fully managed event routing service that allows developers to develop applications using the Maps service to send event notifications to other Azure services that can trigger processes. It is also a cloud service that supports event streaming of Azure resources.

The *publish-subscribe* model of Event Grid can also be integrated with serverless event-driven applications developed with Azure. Webhooks are

valuable if you want to integrate Azure Maps and Event Grid with external services or APIs. As one of the supported features of Event Grid, webhooks allow you to subscribe to specific events and have Event Grid deliver those events to an HTTP endpoint in real time.

When an event occurs, it can send an HTTP POST request to the specified webhook URL containing the event details. The recipient can then process the event and perform proper measures. This provides an easy, adaptable way to integrate Azure services and automate workflows.

For example, you could create a webhook subscription for an Azure Blob Storage container. If you want to get notified every time a new blob is created, the Azure Event Grid sends an event to your webhook that can be programmed to trigger a series of actions such as copying the file to another location, processing the data, or sending a notification.

Overall, webhooks in Event Grid enable you to build event-driven applications and automate workflows across Azure services.

Developing with Azure Maps

It is easy to develop or improve your applications with Azure Maps services. Let's walk through ways to do that.

Develop using REST APIs for the Maps Search service

One of the ways to develop Azure Maps service is by using the Web SDK. The Azure Maps Search service is a RESTful API designed to help web developers search addresses, places, and business listings by name, category, and other geographic data. The service provides information such as geocoordinates, addresses, and place names for mapping and location-based applications.

To use the Maps Search REST API, you must have an Azure Maps subscription and obtain a key. Once you have these, you can create the API URL, send an HTTP GET request, and parse the JSON response to extract the relevant data.

The Maps Search service includes multiple search features:

- Request coordinates for an address (geocode address location) using the **Search Address API**.
- Using the **Fuzzy Search API**, you can search for an address or point of interest (POI).
- Translate coordinate locations into a user-friendly format using the **Search Address Reverse Cross Street API**.

Developing using web and mobile software development kits

Suppose you need to develop your applications on Azure with a feature that requires integration with Azure Maps services other than REST APIs. In that case, there is also support for web and mobile SDKs for different languages and applications developed with Azure.

The following SDKs are available for Azure Maps services:

Maps Web SDKs

The **Azure Maps Web SDK** provides a Map Control client-side JavaScript library. This library enables developers to utilize the functionality and features of embedded Azure Maps and render maps into their websites or applications, for either mobile or web.

Maps iOS SDKs

Maps iOS SDKs are SDKs for the iOS platform that allow mobile or cross-platform developers to easily integrate mapping and location-based functionality into their iOS apps. The iOS SDKs provide tools and libraries that simplify accessing and using Azure's mapping and location services. With minimal effort, developers can add powerful mapping and location functionality.

Maps Android SDKs

Maps Android SDK is a vector map library for the Android platform. With this SDK, developers can add powerful mapping and location functionality to their Android applications with minimal effort.

Azure Maps is designed to be easy to use and integrate into web applications, mobile apps, and IoT devices. Additionally, it is highly scalable, secure, and globally available, making it a reliable choice for organizations looking to build mapping-based applications.

Learn By Doing (Try It!)

The following tutorials are updated based on Microsoft's technical updates for the service.

- [Microsoft Learn Hands-on: Deploy a pre-built module to the Edge device](#)
- [Quickstart: Connect an MXCHIP AZ3166 devkit to IoT Central](#)
- [Tutorial: Connect an IoT Plug and Play module \(C#\)](#)
- [Azure IoT development kit](#)
- [Tutorial: Connect an IoT Plug and Play multiple component device applications running on Linux or Windows to IoT Hub](#)
- [IoT Device Development Documentation](#)
- [Quickstart: Create an Android app with Azure Maps](#)
- [Quickstart: Create an interactive search map with Azure Maps](#)

Summary

In this chapter, you learned the fundamental concepts of the IoT and how it provides practical solutions and innovative technological capabilities. You

learned about Azure’s different IoT solutions and cloud technologies for these IoT solutions. You also learned about the Azure Maps services that enable us to build and integrate geographical map solutions with web and mobile applications. We covered the advantages, disadvantages, and challenges that IoT solutions encounter, how we can handle these challenges, and how to get started building IoT solutions using Microsoft Azure.

Check Your Knowledge

1. After learning about the IoT concepts in this chapter, what are the most valuable benefits of these technologies to us and society?
2. What top three challenges in IoT innovation do we need to solve as developers, engineers, or technologists?
3. Why is building IoT solutions on a cloud platform valuable and efficient?

For answers to these questions, see the [Appendix](#).

Recommended Learning Resources

“AI, Robotics, Data Science, IoT, and Information Engineering.”
Department for Continuing Education, University of Oxford,
<https://oreil.ly/-YbpV>.

“Azure IoT Central REST API Reference.” Microsoft Learn, November 8, 2022, <https://oreil.ly/tEe47>.

Azure Maps Samples, <https://oreil.ly/XY4V9>.

“Evolution of the Internet of Things (IoT).” TechAhead,
<https://oreil.ly/WZwTs>.

“Quickstart: Create an Interactive Search Map with Azure Maps.”
Microsoft Learn, October 11, 2023, <https://oreil.ly/YhWCq>.

Wall, Kim. “Microsoft Defender for IoT Ninja Training.” Microsoft Defender for IoT blog, June 9, 2021, <https://oreil.ly/GARAc>.

- ¹ Tobias Zwingmann, *AI-Powered Business Intelligence*, O'Reilly Media, 2022, <https://learning.oreilly.com/library/view/ai-powered-business-intelligence/9781098111465>
- ² Chase Snyder, “What Is Network Traffic Analysis (NTA)?” Extrahop.com, February 2022, <https://www.extrahop.com/company/blog/2018/what-is-network-traffic-analysis-nta>
- ³ As of the date of writing, only Europe and US geographic endpoints are supported.

Chapter 9. Azure Security, Identity Management, and DevSecOps

As cybersecurity has become a field of rising concern, a growing number and variety of threats and attacks are becoming an everyday reality. In this security battle, attackers are after the lowest hanging fruit, and they need to win only one attack attempt, while defenders need to win every time. Therefore, security is not a group of tasks that defenders must complete; it is not pulling a few levers to raise security posture. Defenders must embrace a holistic approach to security, including and considering everything to succeed and win the battle—architecture, identities, devices, applications, infrastructure, networking, development, lifecycles, operations, management, and, perhaps most importantly, a security mindset. Remember, a conscious security journey traveler must wear many different hats on this voyage that never ends.

—Sasha Kranjac, CEO at Kloudatech, MCT, Microsoft MVP for Microsoft Security and Microsoft Azure, MCT Regional Lead, Microsoft RD, Certified EC-Council Instructor, CompTIA Instructor, and Security Architect

In **Chapter 8**, you learned about developing solutions with IoT and Maps services in Azure. You learned that the challenges of IoT include security, data privacy, and the challenges of building secured IoT applications and solutions. In this chapter, we dive into how to increase awareness about cybersecurity, specifically on the cloud, to develop reliable, robust, and secure Azure applications.

This chapter contains the essential concepts you need to know to secure your existing or new applications and cloud workloads, regardless of whether your IT infrastructure is hosted fully on Azure, hybrid, or multi-cloud.

Cybersecurity and Why It Matters

According to a September 2023 **cybercrime news report**, the car company Ferrari was struck by a **ransomware attack** that exposed their customers' data. This is just one recent example. There are many security threats, each of which has unique characteristics and impacts on those affected. As more companies and businesses consider migrating their applications, data, and IT infrastructure to the cloud, the threats in the cloud have become a significant concern that many organizations and security engineers are trying to protect against.

Figure 9-1 illustrates the security risks that cloud infrastructure, resources, databases, and storage can be prone to.

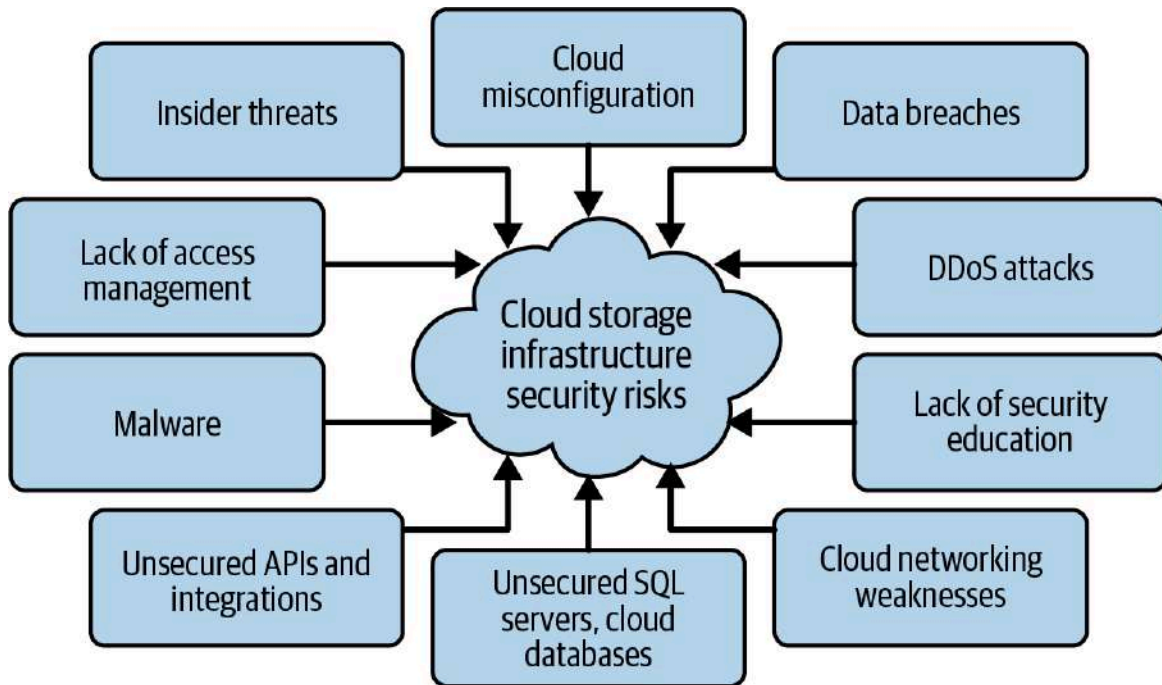


Figure 9-1. Diverse types of cybersecurity risks on cloud storage and infrastructure

The following cybersecurity threats can affect cloud computing infrastructure:

Data breaches

Cloud data breaches occur when hackers get unauthorized access to sensitive applications and user data. Weak

passwords set by the account users are just one of many ways that data breaches can occur. Others include lack of risk awareness, misconfigured permissions, and other vulnerabilities in infrastructure. The **Have I Been Pwned website by Troy Hunt** is a recommended data breach detection website and open source service that will help detect if your e-mail addresses has been affected by a data breach from the web. The **Pwned** book is the result of Hunt's advocating and helping the community and the public in preventing data breaches, and his blog is a good resource for learning more about information security. (Note that this website is powered by Microsoft Azure.)

Distributed denial of service (DDoS)

These attacks can be overwhelming because they target and attack the cloud infrastructure with extreme network traffic, causing the target's system to slow down or become unavailable. Cloud-based DDoS attacks occur when an attacker uses multiple compromised systems to flood a cloud service like a virtual machine or web server cloud with traffic that overwhelms its resources, making it unavailable to the legitimate user. This type of attack can happen in the application layer. Another way is exploiting vulnerabilities in third-party services, for example, on your DNS or NTP servers. DDoS attacks can also be ingested through network protocols. In these instances, the attackers target the weaknesses of your IT infrastructure's network such as public IP addresses, public endpoints, and other system entries. Another form of volumetric attack involves flooding the target with a high volume of web traffic using a botnet. **Figure 9-2** illustrates that hackers who initiate the attack can use these botnets on their target's resources on a cloud infrastructure. These botnet frameworks can attack web

servers, virtual machines, storage, and other resources necessary for the operations of user applications or systems.

Malware

Malware attacks occur in the cloud when infected files are uploaded to your virtual store. For example, computer viruses and malware can be ingested through cloud-based email or messaging services.

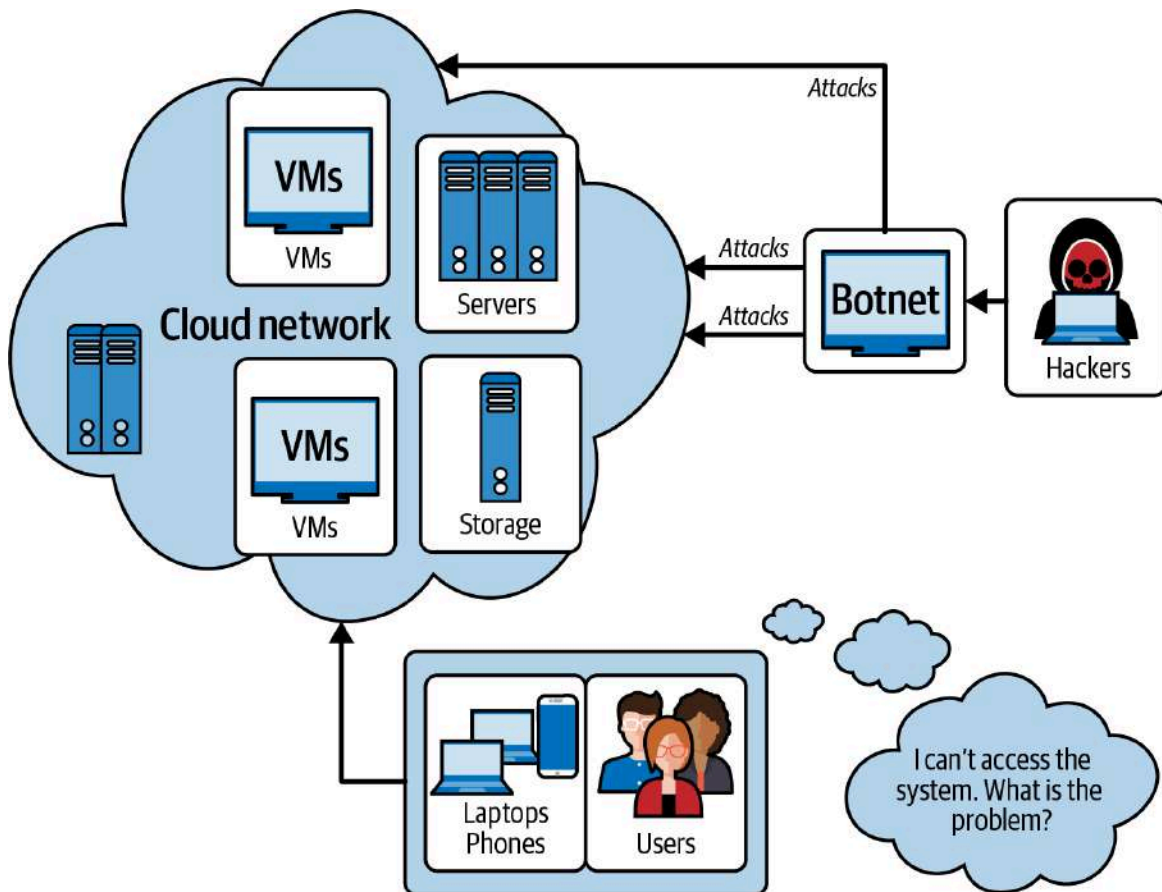


Figure 9-2. DDoS cyberattack on a cloud infrastructure

Insider threats

Issues can also begin within the company. Insider threats can occur when employees or other insiders intentionally or unintentionally cause security breaches. Hazards include employees sharing login credentials, downloading sensitive

data, and misconfiguring cloud security settings. Organizations create security policies and rules to follow, in part, to help prevent these threats from becoming reality.

Misconfiguration of cloud resources

In addition to insider threats intentionally manipulating how systems are configured, misconfiguration can also occur by accident or through a lack of knowledge. The security risks that result from this include misconfigured access controls, storage settings, or network configurations. Cloud resources that can be misconfigured include unsecured storage accounts that are open to the public network through your applications. Misconfigured Azure firewall rules, unprotected Azure VMs, leakage of sensitive credentials from the source code, weak identity management on Microsoft Entra ID, and lack of multi-factor authentication (MFA) configuration all can create security risks. Azure users with owner and global administrator roles, even administrators and security engineers, should be mindful of these. Azure security best practices and patterns is a reference that includes a list of links to **recommended security practices**.

API vulnerabilities

APIs used and integrated with applications and workloads in the cloud can be vulnerable to networking or endpoint attacks if they are not adequately secured. Vulnerabilities and risks related to API integrations in distributed and open source systems can be caused by weak authentication mechanisms, injection attacks, or other API vulnerabilities. One example of this type of vulnerability attack was the **Equifax security breach** in 2017, which compromised the personal data of 147 million people. In this case, the attackers exploited a vulnerability in the Apache Struts web

framework, which Equifax used in their API. Another example is the **Capital One data breach**, which exposed the personal information of its customers.

SQL injection attacks

SQL injection attacks exploit vulnerabilities in web applications to insert malicious SQL code into the application's database. If your databases are not monitored and secure, there is a risk of this kind of attack. SQL injection attacks can steal sensitive information or modify or delete data. Cloud security risks for database servers for Azure SQL, Azure SQL Managed Instance, Azure Cosmos DB, and other cloud databases can arise from misconfiguration, unauthorized exposure of database credentials, access control issues, and other database vulnerabilities. The **OWASP Top 10** is a recommended security standard for developers to follow in web application security.

Man-in-the-middle (MitM) attacks

MitM attacks involve intercepting transmissions between two parties to steal sensitive information or modify data. They can be accomplished by blocking network traffic or imitating one of the parties in contact.

Zero-day exploits

These threats take advantage of vulnerabilities that still need to be discovered or have not been patched by the vendor. These attacks can gain access to systems or steal sensitive information before the vulnerability is discovered. An example exploit is Pegasus, a reported spyware tool capable of infecting mobile phones.

NOTE

Inadequate security awareness and education within an IT organization can increase cloud security risks and cybersecurity attacks. Employees may engage in risky behaviors that leave the organization vulnerable to attacks. Security awareness and education are critical components of a strong cybersecurity posture. By ensuring that employees understand the risks associated with cloud computing and know how to follow best practices for securing cloud resources, organizations can minimize their risk of cyberattacks and data breaches. Learn more about Microsoft's recommendation for cloud security and managing cyber risks at <https://www.aka.ms/securityroles>, which is also part of the Cloud Adoption Framework for Azure.

To protect against cloud cybersecurity threats, enforcing robust protection and implementing policies and standards, such as multi-factor authentication, proper identity management, security, governance, and data encryption are vital. Having these steps in place makes it easier to leverage periodic security audits and vulnerability assessments, which can help pinpoint and manage potential vulnerabilities in cloud infrastructure.

Importance of Cybersecurity on Cloud Infrastructure

According to a recent cloud cybersecurity **survey** by the World Economic Forum, the major security threats today are caused by the misconfiguration of cloud platforms, unauthorized access, and insecure APIs. Cybersecurity protects computer systems, networks, and digital assets from unauthorized access, theft, damage, or disruption. It encompasses many technologies, processes, and practices designed to safeguard information and digital resources against cyberthreats like malware, phishing attacks, data breaches, and other cyberattacks.

So far in this book, you have learned about many technologies that you can use to build and develop solutions. What does that mean when it comes to cybersecurity? Cloud computing has revolutionized how organizations manage their applications and store data, providing on-demand access to shared computing resources over the internet. However, adopting cloud computing has also introduced new security risks, as organizations entrust

their sensitive data to third-party cloud service providers. Many still need proper education, awareness, and strategies to understand and implement secure cloud solutions.

TIP

Cyber Signals is a cyber-threat intelligence initiative of **Microsoft Security** that researches defense against possible security and cybersecurity risks. The reports and analysis provided by Microsoft through Cyber Signals provides your organization with the latest cyberthreat information, which will help you identify and take action on possible cloud security risks within your infrastructure, applications, and resources.

Organizations must implement robust cybersecurity practices and solutions to secure their cloud resources and protect against cyber threats. These include regular security audits, network segmentation, access control, data encryption, and continuous monitoring of cloud environments.

By prioritizing cybersecurity, organizations can mitigate the risks associated with cloud computing and ensure the safe and secure delivery of their services and data to customers and partners.

Effective cybersecurity ensures cloud resources' confidentiality, integrity, and availability. Without proper security measures, cybercriminals can exploit vulnerabilities in cloud-based applications to steal, manipulate, or destroy sensitive data, causing significant financial and reputational damage to organizations.

Zero Trust Methodology in the Cloud

Cloud-hosted applications and workforce mobility have revolutionized how cybersecurity is conceptualized and implemented. On-premises data and applications are shifting to hybrid and cloud-based environments that must be secured.

The new perimeter that organizations now need to secure includes every access point that hosts, stores, or accesses corporate resources and services.

Network firewalls and VPNs may now be bypassed while accessing corporate resources and services. On-premises firewalls and VPNs need more insight, solution integration, and agility to provide timely, end-to-end protection.

As a result, organizations need a new security framework that better adapts to the modern environment, embraces the hybrid or mobile workforce, and secures people, devices, applications, and data wherever they are.

The **Zero Trust** security model assumes breach and verifies each request as if it came from an uncontrolled network. Zero Trust teaches us to *never trust, always verify*, regardless of the request or resource.

Before granting access, a Zero Trust model firmly authenticates, authorizes within policy restrictions, and inspects for anomalies. User identification and an application hosting environment are used to prevent breaches.¹

Micro-segmentation and least privileged access help reduce lateral movement. Finally, deep intelligence and analytics identify what happened, what was compromised, and how to avoid it.

During the authentication process of the Zero Trust model, users need to go through authentication methods to verify their identity as part of the organization. These authentication methods through Microsoft Entra ID can be done by MFA being enforced and other authentication protocols and policies configured by the organization. These policies control which users within the organization can access resources and applications within the Microsoft Entra tenant, for example, authentication and access to certain Azure resources within the subscriptions. These security and access policies help in identifying possible security risks or incidents.

These essential principles of Zero Trust will help you and your organization impose security measures on your Azure resources and infrastructure:

Always verify

Authenticate everyone and approve only verified and permitted users to access your resources. With Microsoft Entra ID and other identity and access management

services, you can quickly implement the verification and authentication required to protect your resources and users using user identification, location, and anomaly detection.

Implement the least privilege principle

Just-in-time, risk-based adaptive policies, and data protection restrict user access to protect data and improve productivity. Applying the principle of **least privilege** in anything in your Azure workloads is a good practice, especially in terms of security and identity management. The feature of Access Reviews in Microsoft Entra ID controls who in your organization gets access to Azure resources.

Assume security breach

Zero Trust means to trust no one, always verify, and always assume possible breaches or attacks in your applications, workloads, or infrastructure. Some ways to minimize breaches include segmenting access by network, user, device, and application. This awareness reduces breach blast radius and lateral movement. Another way to achieve these is by using monitoring and analytics for improvements, detecting threats, and increasing security.

A Zero Trust plan should be implemented across a digital estate and function as an IT organization. It should be an integrated security philosophy, mindset, and process from beginning to end. These practices can be implemented in different security user identities, applications, devices, data, networks, and infrastructure categories.

Planning for adopting a Zero Trust security model is influenced by factors such as different organizational requirements, existing technology implementations, and security phases. Each organization has different maturity levels, and each journey is unique.

Cybersecurity, DevSecOps, and Securing Azure Infrastructure

Organizations are already versed in securing IT infrastructure on premises. Cloud computing brings new security risks that must be carefully managed, and it's important to note that cybersecurity is equally crucial for securing IT infrastructure on the cloud as it was for on premises. Stated another way, it is vital to keep cybersecurity in mind while designing the architecture and developing applications for the cloud. Securing workloads is essential to protect cloud resources and infrastructure from unauthorized access, data breaches, and other security threats.

Personal data like names, email addresses, and contact information may be processed and stored in cloud-hosted applications, making it essential to protect the information from unauthorized access or disclosure. Therefore, encryption, access controls, and threat detection are critical to prevent data breaches.

Organizations that store and process customer data on Azure must keep their customers' trust by safeguarding their data from security breaches. By implementing strong cybersecurity measures in Azure, organizations can display their obligation to protect customer data and prevent security breaches.

Azure's cybersecurity benchmarks, such as the **security pillar** of the Well-Architected Framework for Azure, **Zero Trust security** strategy, and security services will help organizations comply with specific cybersecurity requirements and regulations.

Robust security controls in Azure can mitigate potential security risks in applications, networking, databases, and other resources. Cyberthreats continuously evolve, and attackers always look for new vulnerabilities to exploit. Therefore, constantly implementing strong security measures in Azure can help reduce security breach risks and minimize the impact of any security incidents.

Responsibility for Security Strategies Is a Collaborative Effort

By implementing robust security measures and staying alert for evolving threats, organizations can leverage the full potential of Azure while minimizing security risks. Azure's cybersecurity measures protect sensitive data, comply with regulations, mitigate security risks, and maintain customer trust.

Here are the significant reasons why securing your resources on Azure and genuinely understanding security at all levels is essential:

Shared responsibility model

This model describes the different security responsibilities in cloud deployment models and infrastructure. Cloud providers like Azure offer a collaborative model that is a **shared responsibility**. For example, if you use a PaaS deployment model on Azure, Microsoft, your cloud provider, is responsible for securing the underlying infrastructure. At the same time, customers or users are expected to ensure their data and applications are secure, including their communication and integration with third-party tools and resources.

Increased cybersecurity risk exposure

One of the features of using the cloud is that your applications are portable, more manageable, and accessible from anywhere and any device, which also makes them more vulnerable to cyberattacks. As cloud usage grows, cybercriminals increasingly target cloud environments to steal data or launch attacks. Therefore a robust tool for capturing and identifying these risks before they happen is critical.

Compliance requirements

Many industries, countries, and regions have specific regulations and standards that require organizations to meet cybersecurity, compliance, and data protection requirements. If an organization fails to follow these imposed regulations, their actions can result in substantial monetary penalties and sabotage an organization's reputation.

Cloud data protection

Organizations store vast amounts of data in the cloud, including sensitive customer data and intellectual property. It is essential to take security precautions and measures in your data storage, databases, and the integration services your applications are communicating with.

Business continuity

To continue business operations around the clock is critical in many businesses today. For example, when a security breach or cyberattack occurs, the cloud enables organizations to recover their data and applications quickly and efficiently. However, this requires robust cybersecurity measures to ensure that backups and recovery mechanisms are secure and protected from attack.

Security is crucial to defend data and applications, comply with regulations, prevent cyberattacks, and ensure enterprise continuity. Organizations must enforce vigorous cloud security measures to secure their IT infrastructure and environments and protect them against evolving threats.

Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps

DevSecOps is a methodology incorporating security practices collaboratively within both the **software development lifecycle (SDLC)** and

IT operations. It aims to foster a culture of shared accountability, collaboration, and openness across development, security, and operations teams to develop and deliver secure software products. By focusing on a security mindset, DevSecOps enables organizations to integrate security into all phases of cloud application development, including design, architecture, programming, testing, deployment (CI/CD), and administration.

This approach evolved from DevOps, emphasizing collaboration, automation, and monitoring throughout the SDLC. As the creation of applications has grown more complex, and cyberattacks have increased, the need for security to be incorporated into the methodology known as DevOps has become widely acknowledged. It is beneficial in any cloud application development and engineering since it enables organizations to identify and resolve security issues early in the SDLC.

Additionally, DevSecOps aids organizations in achieving regulatory compliance and mitigating the security risks related to third-party cloud-based services. This approach can assist businesses in decreasing the risk of data breaches, cyberattacks, and other types of security-related events in cloud environments. It is crucial to cloud engineering because it enables organizations to create and deliver secure, high-quality software products in a swiftly expanding technological landscape.

Shift-left and *shift-right* are two approaches to implementing security in the DevOps process, as shown in **Figure 9-3**. Each part of the continuous loop focuses on various stages in the development lifecycle.

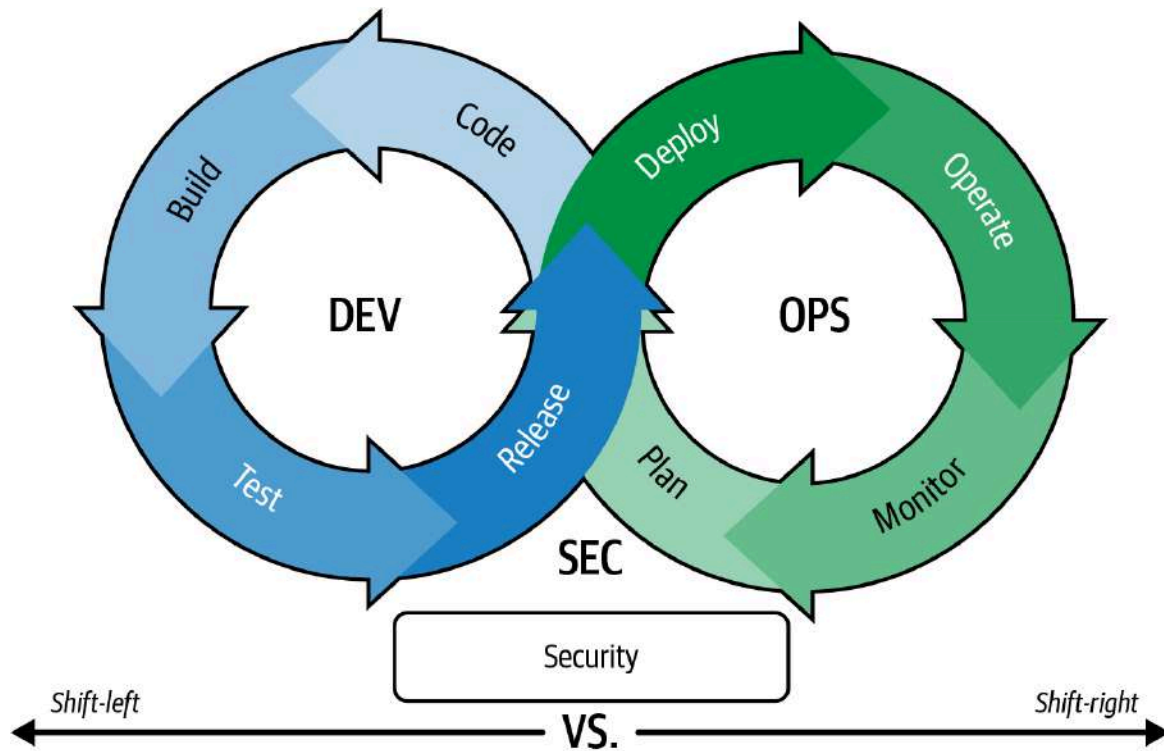


Figure 9-3. Shift-left and shift-right in the DevSecOps continuous loop cycle

Shift-left: Integrating security practices before production

This part of the infinity loop of DevOps focuses on implementing security measures at the beginning of the development and deployment processes, and during the planning and design phases. Shift-left aims to prevent security issues from arising in the first place by identifying and addressing vulnerabilities early during the development phase.

Integrating security measures into the development pipeline as early as possible can include using **automated testing tools** and conducting code reviews to ensure code quality and security. The key benefit of shift-left is that it helps reduce the likelihood of introducing vulnerabilities into the software.

Shift-right: Continuous monitoring and improvement in production

Shift-right in DevOps continuous cycle focuses on the operational phase of the SDLC and implementing measures to identify and quickly respond to

security incidents in production. This approach involves continuously monitoring the production environments and analyzing log data to detect security risk incidents and improvements needed for applications.

On the right side of the cycle, best practices include implementing security monitoring tools and processes, performing regular vulnerability assessments, and developing a robust incident response plan.

Security perspective: Shifting left versus shifting right

Shift-right practices differ from shift-left due to when they occur in the SDLC. Shift-left occurs early, during the development phase, whereas shift-right occurs later during the monitoring or operational phase. But what does this mean in practice?

It means that shift-right practices involve maintaining and monitoring the software in production to improve it. It also means that instead of performing all the testing in the earlier phase of the development process (shift-left), this practice does the testing later in production and takes care of the issues detected as soon as possible.

In shift-left practice, most common unit tests can be done in earlier phases of development and even deployed to an isolated environment like QA and staging. However, some tests can only be performed once the applications are deployed and released to production and have reached their full functionality.

Those who choose to perform testing in production benefit from real-life test cases and a diversity of possible changes and issues that the applications can handle or not handle. Since the production environment is where the actual users are using the application being built, it gives more opportunities to test its durability and reliability. Some common practices used to protect the production environment include testing for consistency and automation and then controlling the changes using ring-based deployments, feature flags, and **chaos engineering**.

Ring-based deployments help DevOps and IT teams implement practices regarding deployments to production workloads. This model is illustrated as

rings of different purposes, types of users, environments, and other essential factors.

For example, if a new feature is to be deployed to production, the early set of users is called canaries from the technique of canary deployment. Canary releases implement small changes or features to a small group of selected users before they go to the production environment with real users.

Once a canary release is successful, the next phase includes rolling it out to selected users called early adopters. They are users or customers who use the feature and are willing to test the release. They are also known as beta testers and may be internal users within an organization.

NOTE

Blue-green deployment entails deploying an update into a different production environment from the current application. After validating the deployment, you switch the traffic routing to the upgraded version. If you are using Azure App Service, you can use the staging slots feature to stage a deployment before deploying it to production.

While there are similarities to canary releases, there are a few important differences. For example, instead of routing all traffic to the improved application, in a canary release, you direct only a portion to the new deployment. If there is an issue, you revert to the initial deployment. If not, progressively redirect more traffic to the updated version. If you're using Azure App Service, you can handle a canary release by testing capabilities in production using Deployment Slots.

If you'd like to deepen your knowledge of the different types of deployments for minimizing risks in production deployments, check out [Martin Fowler's website](#) for more information about canary releases and blue-green deployment.

In Azure, feature flags are easily implemented in Azure App Service. Feature flags can be used to take control of which features of applications are to be enabled or disabled during deployments.

[Azure Chaos Studio](#) is another great tool to use to implement [chaos engineering](#) in software development. This practice tests how durable and resilient your applications are in tolerating failures. Azure's chaos engineering tool will help you and your team by providing tools that enables you to understand and improve your applications to be more resilient.

In summary, shift-left is a proactive approach to security that focuses on preventing security issues from arising in the first place. Shift-right is a reactive method usually focused on detecting and responding to security incidents after they have occurred. Both approaches are essential in securing applications in the DevSecOps process, and they should be used together to provide comprehensive security coverage.

The following section covers the different security services you can use for your Azure workloads and cloud infrastructure to implement and secure your applications.

Azure Security for Applications, Databases, and Networks

Now that you know the foundations of the different cybersecurity threats and have a better understanding of the importance of security in the early application development phases, let's shift gears and talk about the Azure services available to protect your cloud workloads. Azure provides a comprehensive set of security services designed to help protect your data, applications, and infrastructure from potential threats.

In the following sections, you will learn each of these Azure security services and tools spanning categories such as identity and access management, network security, data security, application security, and compliance and governance.

Azure Identity and Access Management (IAM)

Within Azure, every user has an associated identity and permissions that enable them to access and manage resources in Azure based on their role and job responsibilities in the organization. Because of this, IAM is essential to understand the difference between authentication and authorization in identity management in Azure.

Authentication and authorization

While authentication and authorization have related meanings, there are some notable distinctions. Authentication means checking and verifying the identity of a user or service principal and double-checking they are who they are. IAM in Azure provides various authentication mechanisms to ensure that only authenticated users and services can access resources in Azure.

As you learned in **Chapter 3**, Microsoft Entra ID is commonly used for user identity authentication, providing secure access to applications and data. The identity associated with Microsoft Entra ID can be verified using multi-factor authentication (MFA), which adds an extra level of security to accounts by configuring them to require additional authentication factors.

IAM also supports integration with external identity providers, such as social identities or on-premises active directories, to allow external users (considered guests in a tenant) to authenticate with the same credentials they use in their organization.

After a user has been authenticated, authorization checks are initiated. This process establishes what resources a user or service principal can access and what actions they can perform on those resources.

The IAM in Azure provides role-based access control (RBAC), a mechanism for managing authorization that allows you to define roles with specific permissions and assign those roles to users or groups. RBAC simplifies user management by grouping users based on their job functions and providing them with the appropriate access to resources.

IAM is important in Azure because it helps you and your organization protect your cloud resources against unauthorized access from those within or outside your organization. Managing identities and user permissions also helps simplify user management, provides security auditing and compliance features, and enables RBAC and other security management practices.

Microsoft Entra ID security features

Microsoft Entra ID is an identity and access management (IAM) service. As an organization, you can manage the users within your organization's tenant and what these users can access, and control who can access external and internal resources securely with proper authentication and authorization based on the user's allowed permissions.

With Microsoft Entra ID, users can access external resources such as Microsoft 365, the Azure Portal, and thousands of other SaaS applications using a single set of credentials. It also helps control and manage who gets to access internal resources, such as applications on your corporate intranet and any cloud applications developed for your organization, while providing robust security and compliance controls.

Centralizing IAM using Microsoft Entra ID creates ease of user access management and guarantees that your organization allows only permitted users to access your resources.

Microsoft Entra risk detection are services in Microsoft Entra ID that help protect your organization against identity-based attacks and other security threats.

Microsoft Entra ID Protection

Identity Protection is a cloud-based service in the Microsoft Entra ID platform that helps you safeguard your organization's identities by providing risk-based identity protection. It helps detect and remediate risks of identity threats and can also be integrated with other services for better risk monitoring.

It helps analyze data from various sources, such as user sign-ins, and alerts you to suspicious activity or risky behavior. Behind the scenes, it uses ML algorithms to identify these risks from unfamiliar locations or devices.

In addition, it can also provide good security recommendations for addressing the risks that it flags. The service also offers insights and reports to help you understand the risks facing your organization and take appropriate action.

This feature helps you protect your organization's identities against identity-based attacks, such as phishing or credential stuffing. It can promptly mitigate security breaches by detecting and alerting you to suspicious activity. The service also provides insights and recommendations to help you improve your organization's security posture, such as recommending policies to block risky sign-in attempts or requiring multi-factor authentication for high-risk activities.

Microsoft Entra risk detection

Risk detection is a feature within Microsoft Entra ID that uses ML to detect suspicious activities related to user accounts. Examples include user risks such as impossible travel, atypical location, and brute force attacks. The risk detection analyzes data from various sources, such as user sign-ins and authentication logs, to identify and flag suspicious activities for further investigation. This feature also provides insights, alerts, and reports to help you understand the risks facing your organization so you can take appropriate action.

Microsoft Entra Connect for hybrid SSO and authentication

IAM is also possible for hybrid infrastructure. Microsoft Entra Connect enables synchronization between your on-premises Active Directory and Microsoft Entra ID on the cloud. The cloud version of active directory helps facilitate single sign-on (SSO), for example the Microsoft Entra ID applications and services for users. The capability for hybrid authentication operates based on the **Primary Refresh Token (PRT)**.

Once the user's devices are registered with Microsoft Entra ID for hybrid, Microsoft Entra ID can be joined and personal registered devices via work or school accounts can be added. Microsoft Entra Connect is an on-premises solution for hybrid identity scenarios. It is recommended to consider **Microsoft Entra Cloud Sync**.

Hybrid identities on Microsoft Entra ID

Businesses and enterprises are increasingly integrating on-premises and cloud applications. Users need access to be able to work on their

applications, storage, and databases both locally and remotely. This means today's hybrid way of working and business transactions require us to manage users, their permissions, and their identity security both on-premises and in the cloud.

On-premises and cloud-based functionality are both available in Microsoft's identification solutions. These technologies provide a shared user identity for authentication and authorization of all resources, independent of location. These possibilities are made possible by hybrid identity management and Microsoft Entra ID.

One of three authentication techniques can create a hybrid identity with Microsoft Entra ID, depending on the use case. These authentication strategies also offer single-sign-on functionality.

Password hash synchronization

One Microsoft Entra ID feature is password hash synchronization (PHS), a type of synchronization of passwords that reduces the number of passwords users have to remember. It is designed so that the user only has to remember one password.

As illustrated in **Figure 9-4**, users can use the same password to authenticate themselves on both the on-prem Active Directory and Microsoft Entra ID sync on the cloud. The leaked credential detection capability provides additional security for hybrid Microsoft Entra accounts.

PHS reduces the administrative burden and is a good enhancement of the general security posture of Microsoft Entra ID and as an IAM service. Additionally, Microsoft's monitoring and telemetry are used for security benefits, including IP lockout, Smart Lockout, and leaked credentials, and can allow organizations to improve their security management in Azure.

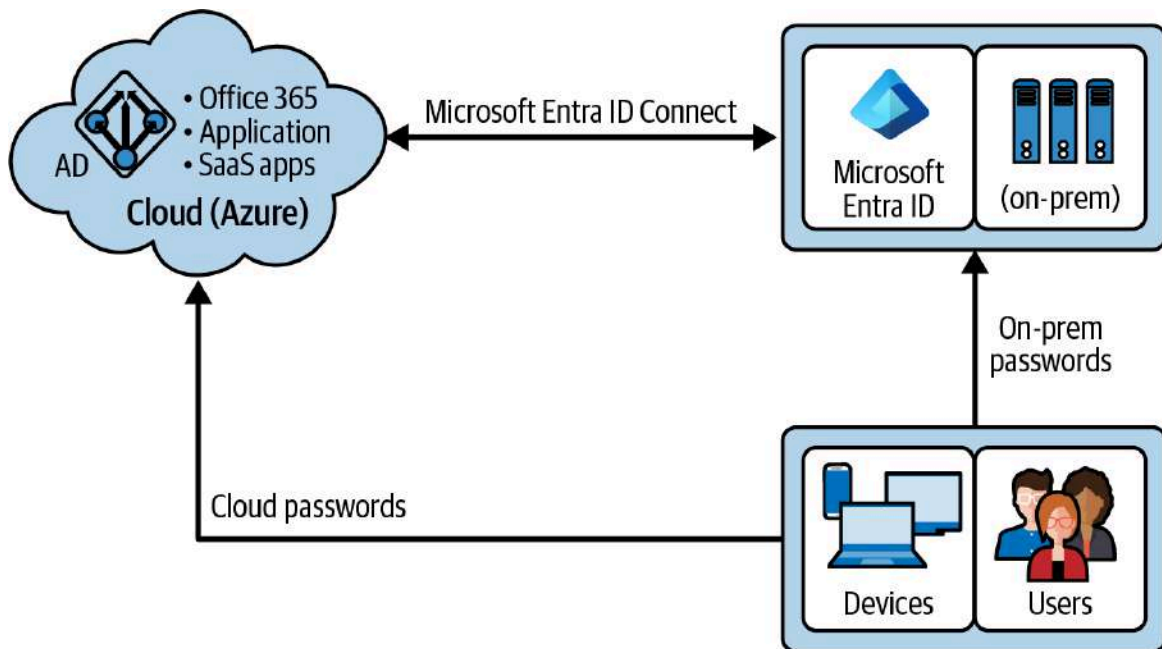


Figure 9-4. Password hash synchronization on Microsoft Entra Connect

So, how does it work? The PHS agent acquires the encrypted envelope and generates a key with the help of the **MD5CryptoServiceProvider** and **password salt** to convert the received data back to the MD4 format. The synchronization agent for password hashes never receives the password in plain text. Avoiding plain text for passwords is a good practice for security purposes.

Pass-through authentication

Pass-through authentication (PTA) for Azure enables your users to authenticate themselves in both on-premises and cloud-based applications using the same password. This feature improves the user experience by reducing the number of passwords they must remember while reducing IT maintenance costs, as users are less likely to forget how to sign in. This feature validates user credentials directly against your on-premises Active Directory when users sign in with Microsoft Entra ID.

As an option to Microsoft Entra password hash synchronization, Microsoft provides organizations with the same cloud authentication benefits. However, if the use case requires that an organization enforce their Microsoft Entra ID security and password policies on premises without the

synchronization to the cloud, they may opt to use pass-through authentication. Using PTA will avoid unrestricted network access to its domain controllers. All network traffic is encrypted, and only authentication requests that are whitelisted and allowed will be permitted.

Active Directory Federation Services

Active Directory Federation Services (AD FS) is another trusted system for authentication that can fulfill advanced authentication requirements. It can serve as an authentication method for Microsoft Entra hybrid identity solutions. When you select this authentication method, Microsoft Entra ID transfers the authentication process to AD FS for password validation. AD FS can satisfy sophisticated authentication requirements such as smartcard and third-party MFA.

AD FS is a Windows Server operating system component incorporated into Windows Server 2008 and later. It enables users to log in to their applications using SSO throughout their organizational boundaries. A single set of credentials authenticates users to multiple applications.

AD FS is also used to allow authenticated users to access cloud-based resources from outside the organization's network in hybrid identity management. It allows users to securely access cloud resources without repeatedly inputting their credentials and is a reliable authentication mechanism that can perform advanced authentication tasks.

Authentication methods include smartcards and third-party MFA. MFA transfers the authentication process to a different trustworthy authentication system, such as **on-premises AD FS**, to validate the user's password.

If you are currently using AD FS, Microsoft highly recommends upgrading to Microsoft Entra ID. This **AD FS to Microsoft Entra ID migration guide** that describes the different migrations is useful.

Azure role-based access control (RBAC)

In the earlier chapter about Azure fundamentals, you learned about Azure RBAC. It is an authorization system developed on Azure Resource Manager

that provides Azure resource access management at a granular level. As an owner or administrator of your Azure subscriptions, Azure tenants, and Azure management groups, it lets you control who has access to Azure assets, what they can do with them, and which areas they can access.

As of this writing, Azure RBAC has over 120 **predefined roles** that can be assigned to individuals, groups, service principals, or managed identities to control access to Azure resources. Azure custom roles can be created if the predefined functions or RBAC roles do not meet an organization's requirements. As for Azure RBAC limitations, you are allowed to have 4,000 Azure role assignments per Azure subscriptions and 500 role assignments per management group. If you are consider to create customized roles, you can create a maximum of 5,000 roles per Microsoft Entra ID tenant.

With RBAC in Azure, you, as an organization or administrator with appropriate roles, can grant users the exact access required to perform tasks. You can divide responsibilities within your team and give users access to perform their jobs. Access to Azure resources is controlled through the designation of different roles within the organization:

- Granular access management for Azure resources based on roles
- Controlling who has authorization and permissions over Azure resources within your team or organization
- Capability to grant or deny permissions at various scopes within the subscriptions, tenants, management groups, resource groups, and even at the resource levels on Azure

Generally, Azure RBAC built-in roles are useful to get started in managing access to your resources and enforce rules based on roles assigned to users and team members in your organization.

Multi-Factor Authentication

Multi-factor authentication (MFA) is a safety measure that requires users to submit multiple authentication methods before accessing their accounts in

an effort to authenticate themselves. These authentication methods may include an SMS code or a biometric aspect, such as a fingerprint or facial recognition, as an alternative to the usual log in and password. Mobile calls, text messages, mobile app notifications, and FIDO security keys are the authentication methods Microsoft Entra MFA supports.

Microsoft Entra MFA is Microsoft's approach to implementing MFA. Users must submit at least two forms of identification using the MFA process before accessing protected assets on Azure, such as applications or data.

The Microsoft Entra MFA methods work as follows:

1. The user attempts to use a Microsoft Entra MFA-protected resource.
2. It requests a second form of authentication, such as a code given to the user's phone or a biometric factor.
3. The user provides the authentication factor.
4. It validates the verification factor and allows access to the resource.

NOTE

Users can choose the authentication methods they prefer, and additional methods can be configured for added flexibility and redundancy.

Developers can also use Microsoft Entra ID libraries and APIs for integrating Microsoft Entra MFA into their apps. These frameworks and APIs simplify integrating MFA into an application without requiring substantial code or custom development.

Microsoft Entra MFA is regarded as a cloud safety best practice for cloud development in Azure. It helps lessen the risk of information breaches and cybercrimes by forcing users to submit several forms of authentication, to help prevent unauthorized access.

Additionally, many organizations may have compliance requirements to meet either with the government(s) where their business operates or with

partner organizations, for example the Health Insurance Portability and Accountability Act (HIPAA) and Payment Card Industry Data Security Standard (PCI DSS). Microsoft Entra MFA can assist developers in fulfilling these legal requirements by adding an extra layer of protection, which lowers the likelihood of information breaches and cyberattacks.

Conditional Access policies (CAP)

CAP in Microsoft Entra ID is an if-then statement that combines signals for decision-making and policy enforcement. It is central to identity-based controls that can be used to implement the necessary access controls to maintain an organization's security.

For instance, a CAP may require that users perform MFA to access a resource. Multiple CAPs can apply to a single user. If more than one CAP is used, then all applicable policies are required for a user to gain access.

CAP is enforced by following the completion of initial authentication. CAP integrates signals for decision-making and the implementation of organizational policies.

Every policy is implemented in two stages:

Phase 1: Collect session details

Gather session information, such as network location and device identification. This information is required for policy evaluation.

Phase 2: Implementation

Use the session information gathered in Phase 1 to identify any requirements. If a policy is configured to block access, the block grant control prevents further enforcement and stops the user. The user is then prompted to complete the remaining grant control requirements met during this initial phase. When the policy is met, then the MFA for hybrid Microsoft Entra ID-joined devices has to be marked as

compliant. Additionally, the approved client application, app security policy, change password, terms of service, and custom controls will be marked compliant as well. Then, apply session rules for App Enforced restrictions, Microsoft Defender for Cloud Apps, and configurable **token lifetimes** once all grant controls are satisfied.

In addition to these two stages, consider adding Conditional Access policies for IAM and security on Azure. You can prohibit or allow access with specific conditions such as adding MFA, device verification, and custom security to verify if the client app access is approved.

If your organization has a requirement to enable Conditional Access policies, the following steps are recommended:

1. Sign in as admin to the **Microsoft Intune Admin Center**
2. Once logged in as an Administrator, Select *Endpoint Security - Conditional Access*, then click *Policies* and select *New Policy*.
3. Provide a name for the policy and select the applicable users, groups, applications, and conditions. *Note*: If accessing via Microsoft Entra ID, select *Security, Conditional Access*, and finally, *New Policy*.
4. Set the policy to report-only mode, if desired, to evaluate its impact before enabling it.

By setting up Conditional Access policies, you can control who can access your resources and under what conditions. CAP can assist with data protection and compliance with security standards and regulations.

To use CAP effectively, the following are the recommended best practices:

Plan your CAP strategy

Determine your organization's objectives, security requirements, user categories, and scenarios. Define clear and consistent policies, align with your objectives, and prevent conflicts or gaps.

Test your CAP policies before implementation

Use the report-only mode or the what-if tool to assess the impact of your policies and identify any errors. Monitor policy outcomes and user feedback and modify policies as necessary.

Regularly assess, improve, and revise your CAP policies

As your business requirements and security environment evolve, you should check your CAP policies to ensure they remain pertinent and applicable. You should also monitor any modifications to the CAP's features and capabilities and use them to strengthen your security posture.

These policies defend and secure Azure resources by enforcing organizational policies based on user, device, and location signals. The approaches determine the risk level for each access attempt based on real-time risk intelligence data. They can restrict the user's session, block access, and require MFA. To get started with the basics, use the [Microsoft Entra Conditional Access templates](#).

Managed Identities on Azure

Managed Identities are a feature of Microsoft Entra ID that help simplify the administration of credentials for Azure-deployed applications and other Azure resources. This identity resource helps eliminate the need for developers to manage and store their certificates or keys, thereby reducing the risk of security breaches and making access control simpler to manage.

There are two Azure Managed Identities types:

System-assigned

These are automatically created by Azure when you deploy specific Azure resources like virtual machines, Azure Functions, and others. These identities are tied to the

lifecycle of the Azure resource itself. This means that when the resource is deleted or deprovisioned, that resource is also deleted automatically.

User-assigned

This type of identity is created and managed separately from the resources they use. They can be assigned to one or more Azure resources, such as virtual machines or Azure Functions, and reused across different resources in the same Azure subscription. The advantages of using user-assigned Managed Identities instead of system-assigned is that you have the flexibility to reuse this identity as an object associated with and authorized in other Azure services and resources.

Using Managed Identities provides improved security by eliminating the need for developers to manage and store their credentials for Azure resources and even Azure DevOps service connections used for deployments. It also helps to keep the access control management more controlled yet simple. Furthermore, by using Managed Identities, you don't need to worry about maintaining and rotating secrets or certificates, which can reduce maintenance overhead. Furthermore, you can also configure your Azure DevOps service connection to Azure using **Workload Identity Federation** through OpenID Connect and Microsoft Entra ID managed identities to prevent the manual process of renewing tokens and keys.

Additional practical use cases for Azure Managed Identities when integrating with other Azure services and DevOps deployment pipelines include:

Authentication to Azure services

Managed Identities can be used to authenticate to Azure services including Azure Storage Account, Azure Key Vault, Azure SQL Database, and many others.

Virtual machine access control

These identities can be used to control access to VMs in Azure instead of requiring developers to manage and store their VM administrator credentials.

Identity for service connections on Azure DevOps

Instead of using the traditional Microsoft Entra ID client tokens that must be renewed often, the Managed Identities can be used for service connections between Azure DevOps for CI/CD pipelines.

Managed Identities are **Service Principals**, which are objects of identity on Azure that can be used for applications, services, and integration in order to communicate with and access resources.

Managed Identities are designed to provide an automated way to manage credentials for applications and services deployed in Azure. These identities for Azure resources are a helpful alternative instead of using **Personal Access Tokens (PAT)** for Azure DevOps or keeping track of the expiration of Service Principal client tokens.

On the other hand, Service Principals are like user accounts used to authenticate Azure resources and services. Service Principals are typically used when registering an application or enterprise applications through Microsoft Entra ID on Azure, when you need to grant an application access to Azure resources. They are also used in creating Azure DevOps service connections to manage and deploy Azure resources during deployments.

For example, a developer must deploy an application or service to access Azure resources like **Azure Resource Manager** or Microsoft Entra ID. Service Principals can be assigned roles and permissions to control access to Azure resources and can also be used to authenticate to external services like **Microsoft Graph API**.

Overall, Managed Identities provide an easy and secure way to manage application and service credentials in Azure, helping to reduce the risk of security breaches and simplifying access control management.

Azure Key Vault

Azure Key Vault enables developers to store and oversee cryptographic keys, secrets, and certificates for application authentication and encryption.

It also provides a safe location for storing and managing cryptographic keys and secrets instead of manually configuring passwords, secrets, and sensitive information hard-coded in applications. It supports various key types, including client certificates and secret keys that are symmetric and asymmetric.

There are numerous use cases for integrating Azure Key Vault with Azure applications. Here are some examples:

Authentication to Azure services

Developers or Microsoft Entra ID administrators can securely store and manage application credentials including connection strings, API keys, and passwords to access Azure resources. By preserving these secrets in this vault, developers can programmatically access them without exposing them to unauthorized parties. Since this is not the best security practice, if possible, you may want to consider using system-assigned Managed Identities.

Storage for SSL/TLS certificates

Azure Key Vault can be used to store and manage SSL/TLS certificates to secure HTTPS endpoints. Key Vault enables developers to securely store credentials and configure Azure applications to use certificates stored in Azure Key Vault.

Store and encrypt data at rest

Developers can administer the encryption and decryption keys for data stored in Azure Storage, guaranteeing the data is secure even if its storage location is breached.

Authentication and authorization

Azure Key Vault can be utilized for managing and safeguarding confidential client information, such as OAuth 2.0 client secrets, used in Microsoft Entra ID-integrated apps for identification and validation. By preserving client secrets in Key Vault, engineers can programmatically access them without exposing them to unauthorized individuals.

Key management for encryption

Management of encryption keys for encrypting data in motion or at rest within their applications is also a feature of Key Vault. To protect sensitive information and restrict access to only those who need it, developers can use it to create and store symmetric and asymmetric keys for data encryption and decryption.

Azure Key Vault provides a high level of security for Azure resources by providing access control, key rotation features, monitoring and auditing, and linking with other Azure services.

Azure Network Security

Azure Virtual Network (VNet) provides features that help you defend your Azure applications and infrastructure by isolating your resources from the internet and other systems and creating a secure network environment.

Here are several ways you can protect your Azure applications and infrastructure using VNet:

Network security groups (NSGs)

NSGs filter network traffic between Azure resources in a NIC or subnet. Using NSGs, you can construct rules that permit or prohibit traffic based on the source IP location, the destination's IP address, port number, and protocol.

Azure Firewall

Azure Firewall is a managed firewall service that protects your VNet network. It enables the creation of rules that permit or prohibit traffic based on source and destination IP addresses, port numbers, and protocol. It also provides the security feature of L3-L7 filtering.²

Azure DDoS Protection

This service defends against distributed denial-of-service (DDoS) attacks. It detects and mitigates DDoS attacks automatically on your VNet.

Azure Bastion

This Azure networking service enables secure remote access to virtual machines (VMs) within an Azure VNet. Directly from the Azure Portal, you can connect to your VMs using RDP (Remote Desktop Protocol), Secure Shell, or SSH over SSL.

Additionally, you can utilize Virtual Network Service Endpoints to link the private IP address space and identity of your VNet to Azure services via a direct connection.

TIP

As a secure connection over the internet, a virtual private network (VPN) between two networks or devices enables you to access remote network resources as if you had a direct link to the remote network. VPNs are frequently employed to securely connect remote employees to their employer's network.

On the other hand, an NSG is a security component in Azure that enables you to filter network traffic between Azure network instances and Azure resources. NSGs contain security rules that permit or prohibit inbound network traffic to or outbound network traffic from multiple categories of Azure resources, including VMs, subnets, and network interfaces.

To learn more about the Azure Networking concepts and recommended practices, check out this Microsoft [documentation](#).

Azure offers multiple security tools for shielding your IT infrastructure. Azure Firewall can filter traffic from networks and secure the network's virtual resources. It is a cloud-based managed network protection service that safeguards Azure VNet resources. It is also a stateful security system with high availability and unlimited cloud scalability. Azure DDoS Protection can also defend your applications and assets from DDoS attacks with constant surveillance and automatic network attack mitigation.

Microsoft Sentinel

Microsoft Sentinel is a cloud-native solution for security information and event management (SIEM) and security orchestration automated response (SOAR). You can find it in the Security services category on the Azure Portal along with other services for Microsoft Defender. It offers intelligent security data analytics and threat intelligence across an organization's enterprise, including the cloud, on-premises, and hybrid environments.

Sentinel collects data from various sources, including Azure activity logs, Azure Security Center, and numerous other Microsoft and third-party solutions. After that, it uses ML algorithms and analytics to discover and alert to potential security dangers, helping enterprises respond rapidly to security problems.

Benefits of using Microsoft Sentinel include:

Management of security incidents

Security incidents can be monitored and managed throughout the entire enterprise, which provides a consolidated platform for security management as part of Microsoft Sentinel's centralized security management capabilities.

Intelligent security analytics

AI algorithms and analytics determine and alert about potential security risks. This enables enterprises to detect and respond to attacks quickly.

Integration with Microsoft and external services

Can be well integrated with Microsoft and third-party solutions, enabling businesses to gather information from multiple sources and make the most of their existing security investments.

Flexibility

As a cloud-native solution, Sentinel offers flexibility and scalability to businesses of all sizes, which is an advantage because it can adapt to current business requirements, workloads, and other essential security requirements.

Cost-effective

Sentinel uses a pay-as-you-go pricing approach, making it a more cost-effective option for companies, particularly as they scale or navigate product usage shifts.

Microsoft Sentinel is a solution for fully integrated and sophisticated security management for businesses. This solution assists companies in

locating and responding to security issues that occur across their entire operation. It is important to consider the pricing considerations by checking the Azure Pricing Calculator and the official documentation about this service when enabling these services.

Microsoft Defender for Cloud

Defender for Cloud is a cloud-native security solution that protects Azure resources, workloads, and other cloud service providers. It provides advanced threat protection and proactive defense against attacks by leveraging Microsoft's threat intelligence and security expertise.

Defender for Cloud can complement and enhance the security capabilities of other Azure security services such as Azure Security Center, Microsoft Sentinel, and Microsoft Entra Enhanced Threat Protection. Defender for Cloud's Defender for DevOps, part of this security platform, enables security teams to secure multi-pipeline environments.

It provides a comprehensive view of the security posture of your Azure resources, including VMs, containers, databases, and more. It can help you identify and remediate security issues before they become significant problems.

Capabilities and features of Defender for Cloud include:

- Endpoint detection and response (EDR) for Azure virtual machines
- Advanced threat detection and answer for Azure resources
- Automated threat response and remediation
- ML-based threat analytics and intelligence
- Integration with other Azure security services

Discussing the details of all of these is beyond the book's scope; however, those that might be useful for cloud development in Azure will be covered in the following sections.

TIP

Check out the recommended learning resources at the end of this chapter and *Microsoft Defender for Cloud* by Yuri Diogenes and Tom Janetscheck (Pearson, 2022) to learn more about the advanced details of the Microsoft Defender for Cloud services in Azure.

Microsoft Defender for Endpoint

Defender for Endpoint is a business-grade security endpoint platform designed to prevent, detect, investigate, and respond to advanced threats on enterprise networks. Its foundational capabilities include competitive anti-malware, attack surface reduction, and device-based conditional access. It also provides centralized management and unified security solutions. Microsoft Defender for Endpoint's managed threat-hunting service is designed to be proactive and helps the Security Operation Centers (SOCs) to identify and respond quickly. It uses roles that are native to Microsoft Entra ID.

Microsoft recommends granting users only the permissions they need to complete their duties. Licenses can be assigned using fundamental permissions management or RBAC.

Uses for Microsoft Defender for Endpoint include:

Endpoint behavioral sensors

Embedded sensors collect and process behavioral signals from devices before transmitting them to your private, isolated, and reliable cloud tenant.

Cloud security intelligence

Uses big data, device learning, and Microsoft's access across the Windows ecosystem to provide valuable insights into the threats facing your enterprise.

Threat intelligence

Recognizes the attacker's methods and operations and produces alerts when these characteristics are observed in collected sensor data. Using ML, it autonomously investigates signals and remediates complex threats within minutes.

Defender for Endpoint is an essential aspect of the assume breach concept of the Zero Trust principle and a key component of your enhanced detection and response (XDR) deployment with **Microsoft 365 Defender**.

Microsoft Defender for DevOps

Many enterprises use Azure DevOps for their deployments. To implement security in DevOps, adding security checks using Microsoft Defender for DevOps helps protect your infrastructure and CI/CD pipelines. Using a central console, Defender for DevOps enables security teams to secure applications and resources from code to the cloud in multi-pipeline environments, such as GitHub and Azure DevOps. The insights and data from Defender for DevOps can be combined with other relevant cloud security insights to prioritize code remediation.

Features of Defender for DevOps include:

- Security administrators have complete visibility into the DevOps environment and its current pre-production code security posture. This includes code, secret, and open-source dependency vulnerability assessment results.
- You can improve the security for infrastructure as code (IaC) and container images to prevent cloud misconfigurations from reaching production environments. This helps security administrators focus on evolving threats of critical importance.

Overall, Microsoft Defender for DevOps is worth considering, especially if you do not yet have any existing security tools implemented in your DevOps pipelines.

Microsoft Defender for Containers

Defender for Containers is the cloud-native solution that improves, monitors, and maintains the security of your clusters and containers, including their applications.

It helps secure your Kubernetes clusters, regardless of whether they operate on Amazon EKS, Azure Kubernetes Service, or Kubernetes hosted on-premises or using IaaS. This defender performs continuous scans of clusters to deliver visibility of container configuration errors and guidance that can assist in mitigating threats that have been found.

For example, for images stored in a container registry like Azure Container Registry, the tools Microsoft Defender for Containers provides can help detect vulnerabilities and make assessment and supervision easier. Threat prevention for clusters and Linux nodes creates alerts for suspicious actions. This type of protection is known as run-time threat protection.

Enable this **security for your containers** on your workloads depending on where you are currently running your clusters for Kubernetes, e.g., AKS (Azure) on-premises/IaaS using **Azure Arc**, EKS (AWS), and GKE (GCP). You will need to follow specific steps including some prerequisite requirements you must adhere to.

Microsoft Defender for App Service

Another option for protecting web-based applications hosted on Azure App Services is Defender for App Service. It identifies and blocks known and undiscovered malware, vulnerabilities, and other security issues in web apps, which enables it to deliver advanced threat protection for those applications.

Operations included in Defender for App Service are:

Endpoint protection

The Defender for App Service offers protection at the endpoint for web applications hosted on Azure App Service, preventing attacks against those web apps.

Real-time threat detection

Detects and responds to security risks in real time, which avoids the occurrence of security incidents and minimizes the effect of any incidents that do occur.

Assessment of vulnerabilities

This defender can perform vulnerability inspections on web applications. These assessments assist in locating and resolving security concerns promptly before the vulnerabilities are exploited.

Microsoft Defender for App Service offers a security solution for web applications hosted on Azure App Service and assists organizations in constructing and deploying secure and compliant web apps for their businesses.

Overall, Azure services for security bring comprehensive security for infrastructure management and resources hosted on Azure. Microsoft Sentinel provides intelligent security analytics throughout the entire company's attack surface. Defender for Cloud services on Azure are recommended if your organization needs security capabilities and advanced protection against threats for applications, resources, and workloads operating in Azure, on premises, and in other clouds.

Security Best Practices for Azure

Secure coding practices are a set of principles and guidelines developers follow to build secure software applications. In the context of Azure applications, secure coding practices are essential for ensuring that the applications are protected against various security threats, such as data breaches, denial-of-service attacks, and malware infections.

The following sections discuss critical secure coding practices that developers need to know when building Azure applications.

Application Data Input Validation

To prevent an application from processing malicious input, developers should validate all user input. This includes validating data types, lengths, and formats to ensure feedback is secure and free of malicious code or commands.

Consider an Azure application that enables users to input data. In such a case, the developer must employ input validation techniques to ensure the data is secure and contains no destructive code or commands. Failure to validate input can lead to SQL injection, cross-site scripting (XSS), and other security threats.

Implementing Security Scanning and Checks in Source Code and CI/CD Pipelines

Developers, DevOps engineers, and platform engineering teams should ensure Azure applications implement proper authentication and authorization processes and management to avoid unauthorized access to sensitive data or functions. This includes using strong passwords, MFA, and RBAC in Microsoft Entra ID.

For example, suppose an Azure application requires users to log in. In that case, the developer should ensure the login process is secure, and the user's credentials are validated against a trusted source.

Another way to protect the application is by using remote management in a secured vault such as Azure Key Vault. Sensitive API keys, connection strings, tokens, and other credentials can be saved and retrieved from the encrypted and secured vault. The CI/CD and automated testing pipelines should be integrated with security and code scanning tools compatible with your current system and setup.

Secure Communication and Integration Between Applications and APIs

Developers and team members responsible for the security of applications and workloads should ensure that Azure applications use secure communication protocols, such as HTTPS, to protect data in transit. For example, if an Azure application communicates with other services or resources over the internet, the developer should ensure that the communication is encrypted and uses secure protocols. Failure to use secure communications can result in data leakage and interception by attackers.

Taking Error Handling Seriously: Not Just Debugging but Also Security

Developers should consider implementing proper error-handling mechanisms to prevent attackers from exploiting application vulnerabilities. This includes handling input errors gracefully and avoiding disclosing sensitive information in error messages.

For example, suppose an Azure application encounters an error. In that case, the developer should ensure that the error message does not contain sensitive information that attackers could use to exploit vulnerabilities in the application. Failure to implement proper error handling can result in information disclosure and other security threats.

The importance of secure coding practices for Azure applications cannot be overstated. By following these principles and guidelines, developers can build applications that are less vulnerable to security threats, which protects sensitive data, business processes, and customers. Moreover, certain coding practices can help organizations comply with regulatory requirements and industry standards, such as HIPAA and PCI DSS.

DevSecOps: Security in Development,

DevOps, and Infrastructure

Earlier in this chapter, we introduced DevSecOps when covering the concepts and practices of shift-left and shift-right. DevSecOps integrates security practices into software development and focuses on the shift-left side of the application development cycle. This means that the security approach combines DevOps with security to produce a more secure and effective software development pipeline.

In traditional software development, security is often addressed after the application has been built. This can result in security flaws and bugs that are costly and time-consuming to repair. By incorporating security into the software creation process, DevSecOps seeks to address these problems.

Microsoft Security has conducted research that resulted in the DevOps threat matrix, which is described in their April 2023 blog posts. The article explains that focusing on the security of our application source code before it gets deployed to production and implementing security in DevOps processes, can help prevent possible cyberattacks in the infrastructure:

*At Microsoft, we have conducted extensive research into the techniques that malicious adversaries may use to attack DevOps environments. We categorized these techniques into their related tactics and mapped these into a threat matrix. This mapping aims to help defenders better understand the landscape and possible attacker actions, so defenders are better equipped to defend against each technique and protect DevOps environments.*³

DevOps is a convergence of people, processes, and technology that provides continuous value to development and operations. It becomes the framework that guides the teams to make security a crucial part of all Agile organizational and development processes.

Security integration within DevOps expedites iterations. It also assists developers, architects, and infrastructure engineers in designing, constructing, and delivering quicker, more secure code. Typically, team collaboration improves when teams share a common objective. These

practices unite the development, operations, and security teams and facilitate increased cooperation.

Security management in DevSecOps facilitates the rapid identification of security vulnerabilities by integrating code scanning, vulnerability scanning, and remediation into the deployment cycle, such as in CI/CD pipelines. Identifying and repairing common vulnerabilities and exposures (CVE) can help an organization's security team improve their security proactively. The CVE list also develops a culture of security awareness within the organization and their vendors.⁴

Adopting Security in DevOps Practices

The need for a culture shift is one of the significant challenges organizations experience when striving to adopt a DevSecOps approach or build cybersecurity into an IT software project. Another challenge is the inadequate preparation for the task due to a lack of security knowledge. Another common challenge is the complexity of DevOps tools integrations from different DevOps vendors, which is typical within teams in some organizations.

Organizations should establish programs for internal security training to increase employee security awareness to combat these issues. The development, security, and operations teams can communicate and collaborate more effectively using a shared environment and open discussion.

Following are the fundamental principles that can help teams overcome these challenges:

Team synergy and collaboration

Developers, security, and operations teams must collaborate for DevSecOps to succeed. During development, all teams collaborate to identify and address security issues. This collaboration between teams is only possible if the strategy

and security mindset shift has already happened at the leadership level of the organization.

Automation

Automating manual processes, deployments, infrastructure, and other repetitive processes are integral to DevSecOps, allowing developers to incorporate security testing and analysis into the development process. This helps minimize the time and effort required to identify and address security flaws and risks.

Continuous testing

Testing is essential and means ensuring the application is thoroughly tested throughout development. This enables the early identification of security vulnerabilities, making their remediation simpler and less costly.

Continuous monitoring

Another essential aspect of DevSecOps is ensuring that the application is monitored and tested even after deployment. This enables the continuous detection and remediation of security flaws.

Continuous improvement

Testing and monitoring during production to capture any possible issues and bugs is critical. It is also essential to continuously check if applications have any improvement opportunities.

Security education and mindset

Implementing secure and dependable cloud applications begins with the developers. Educating various project teams on the significance of adopting a security perspective

throughout the software development and delivery phases should remain one of the most essential and fundamental objectives.

Automated security integration

Integrate automated security testing tools to ensure security testing is a core part of application development, whether the apps are fully on Azure, hybrid, or hosted in multiple cloud environments. Azure provides several automated security assessment tools, including Azure Security Center, Microsoft Defender for Cloud, Azure Policy, and other security services that help its users achieve this in DevOps environments such as Azure DevOps, GitHub, and other supported build and deployment tools.

Automation using infrastructure as code (IaC)

These tools orchestrate the creation and deployment of resources for infrastructure, and include Azure Resource Manager (ARM) templates, Azure Bicep, Terraform, and other IaC tools for the cloud.

Continuous integration/continuous deployment (CI/CD)

Automate the build, testing, and deployment process by implementing CI/CD using Azure DevOps or GitHub Actions. This guarantees that any security and compliance tests are performed at each pipeline stage.

NOTE

See [Chapter 11](#) for more DevSecOps-related information such as infrastructure management using infrastructure as code, policy as code, and other techniques.

Executing security in all of the steps in the application development lifecycle in Azure development entails integrating security and compliance into each phase of the development process, from writing code to deployment and monitoring. By doing so, businesses can develop protected, high-quality cloud applications that meet the expectations of their users and conform to industry regulations and standards.

One cannot exaggerate the significance of these practices in security in DevOps and application development. These crucial practices ensure that applications are secure and dependable by integrating security into the development process. Doing this reduces the risk of security vulnerabilities and other issues, allowing businesses to deliver high-quality software more quickly and effectively.

Learn By Doing (Try It!)

The following tutorials are updated based on Microsoft's technical updates for the service.

1. [Quickstart: Microsoft Entra seamless single sign-on \(SSO\)](#)
2. [Learning path on how to secure your data and applications](#)
3. [Introduction to key Azure network security services](#)
4. [Configure Microsoft Defender for Cloud Apps for advanced scenarios](#)
5. [Integrate third-party identity providers with Microsoft Defender for Cloud Apps](#)

Summary

This chapter emphasized the importance of cybersecurity and cloud security in technology, highlighting the need for secure coding, encryption, and authentication in cloud development. It's important that teams genuinely

understand how to ensure cloud resources are secured and managed correctly when using cloud services.

DevSecOps is a recommended mindset that integrates security into software development and identifies vulnerabilities early on. Microsoft Defender for Cloud is a cloud-based security solution that detects and responds to threats, including advanced threat protection and automated incident response.

Additionally, application security, IAM, conditional access, Zero Trust, and MFA are crucial to securing cloud resources. Azure Key Vault is a service that safeguards cryptographic keys, certificates, and other secrets, ensuring the confidentiality and integrity of sensitive data.

Check Your Knowledge

1. What is multi-factor authentication (MFA), and how can it be used with Microsoft Entra ID?
2. What is Microsoft Defender for Cloud, and how does it help protect cloud resources?
3. What are best practices for securing keys and secrets in Azure Key Vault?
4. How can you use Azure Policy to enforce security and compliance requirements in your Azure environment?
5. What is DevSecOps, and how does it differ from traditional DevOps?
6. What is Azure Firewall, and how can it help protect your Azure virtual networks?

For the answers to these questions, see the [Appendix](#).

Recommended Learning Resources

Armitage, Josh. *Cloud Native Security Cookbook*. Sebastopol, CA: O'Reilly Media, 2022.

“Connect Your Azure Subscriptions.” Microsoft Learn, July 10, 2023, https://oreil.ly/xJMj_.

“Cybersecurity Framework.” National Institute of Standards and Technology (NIST), <https://oreil.ly/TESyu>.

Diogenes, Yuri, and Tom Janetscheck. *Microsoft Defender for Cloud*. Microsoft Press, 2022.

Fillingham, Nic, and Natalia Godyla, Security Unlocked, Microsoft Security podcast, <https://oreil.ly/R1xDG>.

Kranjac, Sasha. *Microsoft Defender for Cloud Cookbook*. Birmingham, UK: Packt Publishing, 2022.

“Microsoft Defender for Cloud Documentation.” Microsoft Learn, <https://oreil.ly/gB37e>.

“Microsoft Entra Conditional Access Documentation.” Microsoft Learn, <https://oreil.ly/qxt7Z>.

“Microsoft Entra Managed Identities for Azure Resources Documentation.” Microsoft Learn, <https://oreil.ly/qsYC->.

“Microsoft Security Blog.” Microsoft Security, https://oreil.ly/qPG_L.

Natwick, Dwayne. *Microsoft Identity and Access Administrator Exam Guide*. Birmingham, UK: Packt Publishing, 2022.

“Overview of Defender for DevOps.” Microsoft Learn, July 5, 2023, <https://oreil.ly/DvJ3h>.

“Overview of the Security Pillar.” Microsoft Learn, November 30, 2022, <https://oreil.ly/R6AKO>.

“Security Recommendations – A Reference Guide.” Microsoft Learn, September 27, 2023, <https://oreil.ly/bNWpq>.

Toroman, Mustafa, and Tom Janetscheck. *Mastering Azure Security (Second Edition)*. Birmingham, UK: Packt Publishing, 2022. Wilson, Glenn. *DevSecOps*. Rethink Press, 2020.

- ¹ Rudra Mitra, “Stay compliant and protect sensitive data with Zero Trust security,” Microsoft Security, April 24, 2023, <https://www.microsoft.com/en-us/security/blog/2023/04/24/stay-compliant-and-protect-sensitive-data-with-zero-trust-security>
- ² “Layer 3 vs Layer 7 Firewall: What’s the Difference?” Logix Consulting, September 17, 2020, <https://logixconsulting.com/2020/09/17/layer-3-vs-layer-7-firewall-whats-the-difference>
- ³ Ariel Brukman, “DevOps Threat Matrix,” Microsoft Security, April 6, 2023, <https://www.microsoft.com/en-us/security/blog/2023/04/06/devops-threat-matrix>
- ⁴ CVE List Homepage: A public list for disclosed cybersecurity vulnerabilities, <https://cve.mitre.org/cve>

Part IV. Integration, Infrastructure, and DevSecOps

This fourth part contains chapters focusing on Azure integration services that prepare you to integrate cloud-native applications, including existing ones (on the cloud or on premises). This part also guides on how to develop applications with DevOps tools for cloud computing. You will learn about the importance of DevSecOps to infrastructure and integration and Azure DevOps for team collaboration for developers and IT operations. The upcoming chapters also highlight tools and practices for automation, deployment pipelines, version source control, and Github Actions. You will also learn how to monitor and troubleshoot Azure resources using Azure's monitoring tools.

Chapter 10. Azure Cloud Integration Services and Tools

Modern applications are distributed in the cloud and on-premises, and integrating applications and data is inevitable.

—Ezhilarasi Chezhiyan, Program Manager at AppViewX

Thus far in this book, you've learned about cloud computing and the different services in Azure. The focus has been computing, networking, analyzing, securing, and developing solutions.

In this chapter, we shift gears to cover Azure's different integration tools and cloud services. These integration services will give you the fundamental knowledge and understanding to prepare you for integrating and connecting your applications and services hosted in the cloud. You will also learn how to connect on-premises applications with cloud services using the integration services in Microsoft Azure.

Cloud Integration: An Introduction

Cloud integration links cloud-based applications, networks, and services to establish a seamless, unified environment. It enables sharing of data and applications across multiple platforms and improves businesses' effectiveness, responsiveness, and scalability.

Integration methods on the cloud connect different cloud systems and applications using middleware technologies and APIs. It ensures that their components can communicate and share data. One example is integrating cloud-based apps such as customer relationship management (CRM) and enterprise resource planning (ERP) systems.

Cloud integration can streamline business operations, reduce manual processes, and enhance data precision. It can also improve team

collaboration and data sharing and facilitate improved decision making. Organizations and businesses benefit from improving their operations by integrating systems and applications in the cloud. Modern cloud integration provides reliable, scalable, and faster data synchronization between different systems, regardless of where they are hosted on the cloud or on premises.

Types of Cloud Integration in Azure

There are multiple cloud integration methods, and each varies depending on its purpose, such as point-to-point integration, replication of data, and API-based integration. Internal IT teams or external vendors specializing in cloud integration can perform these. Common cloud integration scenarios include:

Application-to-application integration

In this type of integration, various applications or services are integrated within the organization to improve functionality. This application-to-application integration can be on premises, cloud, or hybrid. Integration involves issuing requests and commands to trigger business events or processes.

Data integration

Integrating data means synchronizing data in different resources. This data is sourced from databases or other data resources and can be transported or processed during the data integration.

Cloud-to-cloud integration

Cloud-to-cloud integration combines two or more cloud-based applications or services to facilitate data sharing and shared functionality. Integration between cloud

environments is possible via APIs, multi-cloud technology solutions, and other middleware technologies.

On-premises to cloud integration

On-premises to cloud integration involves linking legacy or on-premises systems to cloud-based applications or services. Middleware technologies, such as cloud connectors and virtual private networks (VPNs), can accomplish this.

Hybrid cloud integration

Integrating on-premises systems and cloud-based applications or services to create a hybrid environment is the objective of hybrid cloud integration. Integrating hybrid environments with intermediary technologies such as *integration platform as a service* (iPaaS) and hybrid integration platforms is possible.

Cloud-to-mobile

This integration involves linking apps that use the cloud or services with mobile phones and tablets to improve data sharing and functionality. Integration between the cloud and mobile applications is possible via mobile application development frameworks and API-based integration.

Cloud-to-business integration

Cloud-to-business integration connects cloud-based apps or services with business partners or customers to facilitate data sharing and functionality. The cloud to enterprise systems integration can be achieved through various means, such as electronic data interchange (EDI) and web services.

Cloud integrations can also be combined. Aside from the integrations mentioned, there are additional methods of enabling applications and

systems to communicate and share data. These include external APIs, web services, software as a service (SaaS) solutions, business to business (B2B), the IoT, and event-driven and microservices technologies that allow distributed applications to be integrated to build cloud applications.

Benefits of Cloud Integration

Every organization has challenges and problems that integration solutions can solve. For example, most organizations handle time-intensive tasks such as inputting data into two or more systems. This duplication of effort is frequently manual and uses needed resources that could better serve other purposes. On the more severe side, organizations also face challenges such as unexpected system failures or data loss.

These problems and challenges can affect the productivity and efficiency of the organization's workers, and if the systems are in production, they may also affect the business users and customers. This can lead to unhappy users, customers, business partners, or unproductive teams.

The following sections will discuss how well-integrated systems can help with these IT challenges.

Reliability and Scalability of Applications

There's much to say regarding reliability. This book covers the topic briefly, but if you'd like to understand more about the concepts of reliability, scalability, and building robust applications, check out *Designing Data-Intensive Applications* by Martin Kleppmann. He discusses the importance of reliability in our systems and covers other essential topics, such as bugs in business applications that can affect productivity and result in unexpected expenses or damage a company's reputation.

IT systems are often built with the ambitious goal of being reliable and scalable.

Reliability is an application's ability to operate consistently and predictably under different conditions and over time. A reliable application should minimize downtime, errors, and data loss and recover quickly from failures or disruptions, such as outages that affect not only an organization's employees but the entire business.

You need the right technologies, tools, and knowledgeable engineering teams to build a reliable system to plan and implement a cloud integration strategy. These teams must have the organization's business purposes in mind and collaborate well with other teams to ensure that the integration works across the organization.

Scalability is closely related to reliability because a scalable application must be designed to handle current and future workloads as the application grows. A scalable application that is not reliable can cause significant problems for its users and the business, such as lost revenue, damaged reputation, or legal liability.

You can think of scalability as an application's capability to accommodate increasing levels of demand, usage, or growth without degrading its efficiency, availability, or reliability. Scalability is crucial because as an application acquires more users or data, its burden and complexity increase, which can cause it to become sluggish or unreliable, or even crash. A highly scalable application with enhanced architecture can smoothly adjust to these shifts by adding resources to preserve its stability and performance.

Furthermore, other benefits such as backup options for data, high-level security to protect your data, monitoring, disaster recovery, and other observability tools are also part of increasing the reliability of systems.

Most of the cloud providers, including Azure, provide integration tools such as Azure Monitor, Application Insights, and other built-in tools within each Azure resource; however, integrating existing systems to any cloud platform takes time and effort. If planned well and done correctly, data and systems integrations in the cloud can significantly benefit any organization.

For examples, applications hosted on Azure can take advantage of the flexibility of the different kinds of use cases of the integrations mentioned

earlier. For example, you can use Azure Logic Apps if your use case requires you to orchestrate work processes in different applications or systems. Integration implementation with Logic Apps can also be extended with other Azure services for event-driven serverless scenarios, such as integration with Azure Functions, Event Grid, Service Bus, and many more.

In general, developing a scalable and dependable application helps ensure it can adapt to the changing needs of your users and the market, all while delivering a positive user experience and safeguarding your business's interests. To accomplish reliability and scalability, you must consider application architecture, robust cloud infrastructure, reliable cloud services, automation, monitoring tools, and adopting best practices for security, performance, and resilience.

Improved Work Efficiency and Cost Savings

Manual operational tasks in an organization can be time-consuming. Cloud integration eliminates repetitive activities, streamlines workflows, and enhances efficiency by removing the requirement for switching between various apps. This helps save time, decreases errors, reduces labor expenses, and eliminates the need for infrastructure and maintenance on-site. Furthermore, integration of existing systems using cloud technologies eliminates the need for new or custom systems. It allows businesses to save money, remain competitive in the market, and adapt to shifting business and market requirements.

With cloud integration automation, tasks can be done by routines managed by a service. This eliminates user error and reduces the time spent on tasks. As a result, teams can focus on more complex tasks, thus making better use of their time.

Business Agility and Better Business Processes

Integration with the cloud enables businesses to link and combine several cloud-based apps, services, or data sources to produce a unified source of reality for information. It allows organizations to collect, analyze, and

exploit data in real time from various sources, offering them immediate understanding that can help them optimize their processes, uncover new possibilities, and make decisions based on accurate information.

A data-driven approach is significant because it enables organizations to make decisions using the data rather than making assumptions or using their intuition. This results in increased performance, profitability, and competitive advantage.

Exciting research by Christina Delimitrou, discussed in “**Cloud Computing for Agility, Complexity, and Speed**” by J. Edward Anthony, highlights how cloud computing management could be improved using data-driven approaches. She says, “Don’t rely on people to solve this problem. Rely on the data. Rely on the system to tell you how to optimize the system.”

Data-driven approaches with cloud computing are made possible with the help of cloud integration tools creating synergies between cloud integration and business agility.

Cloud Integration on Azure

Azure provides solutions for connecting and integrating services, APIs, and applications that can help your business and engineering teams. Azure has services for API management and other integration tools in different deployment models that provide new opportunities for developing business models. When there are opportunities to create new business models, there are also opportunities to increase business revenue.

Azure also provides solutions that can help produce and secure your applications and your enterprise data without compromising performance, scalability, and reliability.

In the following sections of this chapter, you will learn about the fundamentals of Web APIs, and the different and valuable Azure integration services that will help you design, build, and develop better systems, services, and applications in the cloud or hybrid scenarios.

Introduction to Web APIs

The term “API” in software engineering or cloud development often refers to Web APIs accessed over the internet, like our favorite web applications. These components act as mediators between applications or systems that allow them to communicate with each other. This allows you to have handy things on your smartphone like an app that gives you up-to-date information about the day’s weather.

Different Types of Web APIs

To understand how cloud integration in Azure works, it is helpful to learn about the different types of Web APIs. Public APIs, partner APIs, internal APIs, and composite APIs are specific Web APIs with a variety of purposes:

Public APIs

These are APIs exposed to third-party developers and others in the community. Typically, these APIs enable developers to create applications that can access the services or data of a company. Frequently, public APIs are used to expose social networking platforms, weather information, and travel reservation services.

Partner APIs

These types of APIs are widely used and usually available to selected partners or certain parties that require the implementation of strict rules, often related to security and authentication. Examples of commonly used partner APIs include the Twitter Essential API, which allows its users to access Twitter’s legacy endpoints if you have the API key. Other examples also includes Airbnb API, eBay API, and Microsoft Partner Center Rest APIs.

Internal APIs

Internal APIs share data and services between teams and departments within a corporation or organization. These APIs are not generally available to people or developers from third parties and are predominantly used to enhance internal workflows and communication. Internal APIs are frequently employed to automate business processes, reduce development time, and improve team collaboration.

Composite APIs

Composite APIs combine data or services from multiple sources into a single API. This allows developers to access multiple services with a single API query, thereby reducing complexity and enhancing performance. They are implemented in enterprise applications that require data access from multiple systems or applications.

In conclusion, public APIs are accessible to the general public and external facing. Partner APIs resemble public APIs but are designed for a particular group of partners. Internal APIs are used internally within an organization. Composite APIs aggregate data or services from multiple sources into a single API.

API Management Lifecycle

An API has a lifecycle requiring structure and control management. An effective API management solution should support the entire lifecycle of an API, as shown in **Figure 10-1**, which typically includes planning and initial design, development, testing, deployment, operations, versioning, and retirement.

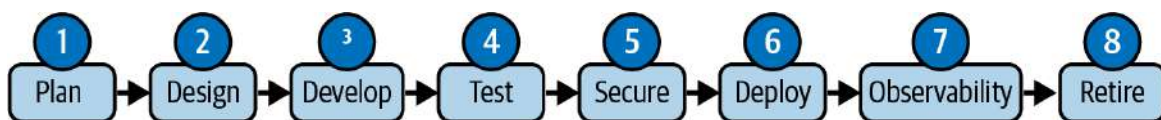


Figure 10-1. API management lifecycle

API lifecycle management benefits businesses or organizations in different ways, including the ability to create, design, develop, and manage APIs for their applications. It enables teams to adopt an API-first strategy and maximize productivity. API lifecycle management uses proven methods to design, build, and deploy APIs. These methods reduce confusion and unnecessary work, improving teamwork and production.

Furthermore, API lifecycle management helps create a roadmap for every API-related project, giving executives more API visibility. It provides the basis for a successful API monitoring plan for monitoring every API's health, efficiency, and consumption.

As teams develop each API, its lifecycle will differ depending on the business use case. The following stages can help organizations and their teams in standardizing their processes, policies, and workflows for APIs developed with cloud applications.

Plan

Developers and API architects plan and implement the API, considering its use, functionality, and performance. They also create a shared workspace and a repository with a CI pipeline. These stages stabilize the API lifespan and set stage-specific locations and tools. The planning stage of this lifecycle helps in preparing and improving API quality before the API is fully developed.

Design

API design needs thoughtful data presentation choices. Users and other programs need to comprehend an API definition's intended functionality. OpenAPI and AsyncAPI define APIs. These specs standardize API definitions, contracts, documentation, mocks, and testing.

Develop

After designing the API, developers build code to implement the functionality. Most development teams utilize Git. They can track changes and securely revert. They use their prepared repository tool to store their source code, track issues, and review work. To standardize their approach, the team must identify software development workflows, which vary widely.

Test

APIs are tested during development and deployment. Developers and QA teams can verify API functionality. CI/CD pipelines can automate or manually test APIs. Early testing helps teams identify and fix issues before they reach production.

Secure

The API lifecycle checks an API for common vulnerabilities that could compromise an application's security. It's crucial to check that an API's authentication logic confines data access and API interaction to authorized users. CI/CD pipelines can manually or automatically perform API security checks to ensure that an application's APIs meet security standards. Access control and encryption can safeguard the API.

Deploy

Deployment means releasing APIs across environments. Many teams use CI/CD pipelines and API gateways to streamline deployment and test and secure updates before releasing them to customers. Agile teams that release new code many times a week need consistent methods to make deployments more predictable.

Observability

In production, API telemetry data is gathered and displayed, and it can be used to create triggers to notify certain users. Site reliability engineers (SREs) and DevOps engineers can automatically configure monitors to inform them of API performance and security issues. They also utilize the contextual API performance data of application performance monitoring (APM) tools. API observability helps identify API errors, delays, and security vulnerabilities before they damage dependent services, partners, and customers.

Retirement

Over time, APIs can become obsolete. When an API gets inefficient, insecure, and unsupported, usually it becomes a candidate for deprecation. The API needs to be deactivated and archived in a controlled, consistent way. To properly and effectively deprecate or retire an API, it is important to communicate this with the end consumers and people that are involved in using it. Providing alternative APIs as a solution helps users deal with the retirement of such API.

Although the stages are defined, every API lifecycle is unique. The most important factor to remember is that teams that follow a well-defined API lifetime tend to be more productive and can produce a high-quality API. A practical API governance approach provides a stable API lifetime, establishing a stable framework. Implementing a governance strategy in the API lifecycle allows you to apply policies and standards that will help you build secure and reliable APIs for your applications.

Azure API Management

Azure API Management (APIM) is a flexible integration service that allows users to design, develop, and implement solutions to connect applications,

APIs, and systems. It is a platform as a service (PaaS) for APIs of different environments. It supports the entire API lifecycle and hybrid scenarios, even multi-cloud use cases.

In terms of pricing options, APIM has different pricing tiers depending on use cases, implementation types with other Azure services, or if it is for non-production or production environments:

Consumption

For lightweight application workloads that needs implementation with serverless solutions this can be a good option, especially if you prefer to pay per use. This tier is not ideal if you are using the cloud for the US Government or a 21Vianet cloud on Azure.

Developer

This pricing option is ideal if you need the APIM service for non-production or development purposes and do not require SLA support.

Basic

Ideal for a small or entry-level production use.

Standard

Ideal for medium or traditional production use.

Premium

Used for large enterprise applications or systems that have high demands for API management. This tier has multi-region support for APIM instances, and it provides the option for you to host your instances on your organization's VNet.

Each APIM tier has different features and capabilities. Check [Microsoft's documentation](#) for information on these and on pricing.

Benefits of Azure API Management

The following are additional benefits of Azure API Management:

- Enables and supports adopting the complete API lifecycle for different use cases in cloud, hybrid, or multi-cloud environments
- Provides an effective developer experience for integration with IoT applications, web or mobile apps, smart watches, wearables, etc.
- Minimizes security risks of exposing APIs hosted in Azure or offered externally to the public
- B2B integration provides options to communicate and exchange business data
- Creates opportunities for collaboration and removes the overhead of point-to-point integration
- Supports compliance and security management of APIs, such as configuring authorization token swaps, certificates, etc.
- Helps in the abstraction and modernization of the backends of legacy applications by enabling them to be accessible by modern cloud apps or APIs
- Provides support for API security using Azure networking services like Azure Private Link
- Allows the flexibility of developing APIs with a [microservices](#) architecture

TIP

If you want to dive deep into the technical details of RESTful Web API design for cloud applications, see Microsoft's [guide](#) for best practices and [recommendations for implementing APIs](#) in your applications.

In addition to management benefits, APIM enable cloud integration's functional capabilities by supporting other Azure resources and services, and it also has [observability features](#) within the Azure platform.

[Table 10-1](#) describes some of the Azure services compatible with Azure API Management, including tools for monitoring, security, IAM, development, and networking.

Table 10-1. Azure resources that are compatible with Azure API Management

Azure Service	Description
Microsoft Entra ID	Developer authentication and authentication of API requests
Azure Monitor	Monitor logs, set up alerts, and perform API management operations and events
Azure Key Vault	Secret management and client certificates for your API
Application Insights	Insights for troubleshooting using end-to-end tracing, live metrics, etc., of Azure API Management
Azure Functions	Serverless compute service to develop event-driven serverless solutions in applications and APIs provisioned in Azure API Management
Azure Web Apps	Host applications in the PaaS deployment model and provide integration features with Azure API Management
Azure Logic Apps	Low-code or no-code tools for automated workflows and orchestrations to complement API management when communicating with APIs, client applications, other backend services, and SaaS platforms
Azure Service	Send traffic to a stateful or stateless web service or

Azure Service	Description
Fabric	backend API
Azure Networking	Network-level protection using Application Gateway, private endpoints, virtual networks, etc.
Azure Service Bus	Enterprise messaging service broker
Azure Event Hubs	Event streaming tasks associate an event stream with your API requests

APIM is a flexible integration service in Azure that allows users to design, develop, and implement solutions to connect applications, APIs, and systems in cloud, hybrid, or multi-cloud environments. It supports organizations and developers in managing the entire API lifecycle.

Components of Azure API Management

Now that you know the benefits of APIM, it's time to learn about components. APIM has three major components: the API gateway, the API management plane, and the developer portal, as shown in [Figure 10-2](#). Currently in preview (as of the time of writing) is another component, [Workspaces](#) for Azure API Management. It allows different development teams to manage their own respective APIs.

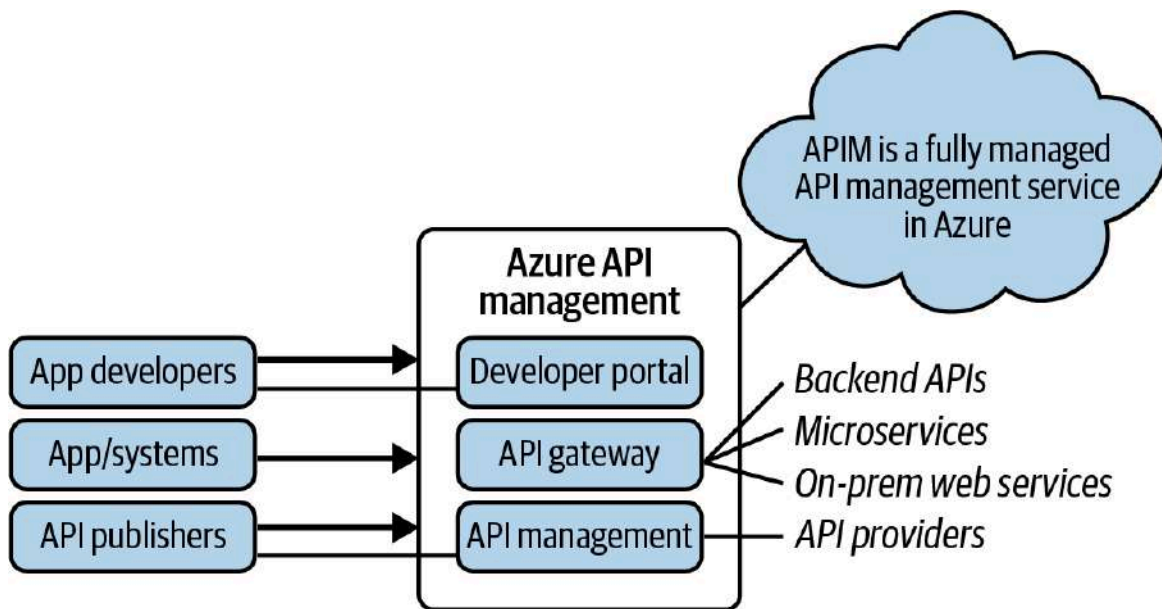


Figure 10-2. Azure API Management service components

Following is information on the functions of these three components and how each can help you design your APIs with APIM.

Azure API gateway

When implementing the APIM service to build and integrate your APIs for your applications, you must first set up the API gateway. The API gateway is the first point of contact for all requests from your client application.

Once the client requests come via the API gateway, it will then forward those requests to the backend APIs. When this communication happens, providers can integrate and communicate with an API's implementations and backend architecture behind the scenes, without affecting API consumers.

The API gateway supports security, routing configuration, caching, throttling, quotas, rate limits, etc. It also allows you to perform security verification of API keys and credentials, such as certificates or JWT tokens, during API operations. You also have the option to configure your API gateway to improve response latency and workloads of backend services by caching the responses.

Azure API management plane

The API management plane allows API providers to comprehensively manage and interact with their APIs. This component can be used to provision and configure the API management service. Furthermore, developers can configure other settings and capabilities, such as packing the APIs as products, managing service users, and, most of all, defining APIs. Suppose you want to set up OpenAPI specifications and integrate your applications with other backend services with your APIs; you can import the schemas and manage monitoring, networking, and analytics using the API management plane.

Using the Azure Portal, Azure CLI, Azure PowerShell, and supported SDKs for common programming languages, you can use the API management plane to import schemas and manage monitoring, networking, and analytics.

Azure APIM developer portal

If you work with application development in the cloud in Microsoft Azure, you must work with integration services and development tools to use APIs. The Azure API Management service provides a portal for developers to design, build, and test their APIs with their client or backend applications.

The developer portal is an open source website that allows you to customize it according to your API and your company's branding or consumer content preferences. API providers can use the portal to call or interact with an API using the APIM console. You can also create accounts, manage API keys, read the API documentation, set up personalized analytics, download API definitions, etc.

These API management components make integration of applications with APIs easier to design, develop, implement, and manage on Azure's platform.

TIP

If you want to dive deeper into these components' technical details, read Microsoft's official [documentation](#).

Azure Logic Apps

In [Chapter 3](#), you learned about Azure Functions for creating serverless workflows programmatically by code. Azure Logic Apps also enables you to author workflows in a low-code way.

Azure Logic Apps provides many integration capabilities and workflow features to help you and your organization design, orchestrate, and automate your business processes. Azure Logic Apps enable users with no programming experience to create automated integrations and workflows with the Azure Logic Apps visual designer.

The web-based workflow of the visual designer allows you to create integration workflows and orchestration for your data, systems, and applications. You can also use it for B2B and enterprise application integration (EAI).

Benefits and Uses of Azure Logic Apps

Azure Logic Apps offers low-code connectivity, a wide selection of connectors, a stable and flexible platform, and monitoring and management. These features ease integration procedures and automate corporate processes, enhancing productivity.

The following are some of the possible use cases for Azure Logic Apps for integration with your existing or new applications:

- Schedule or create workflows for sending emails when someone uploads a file to Azure Storage or new data saved into a database

- Automate business processes from on-premises applications to the cloud by using Logic Apps to help with order management processing
- Monitor events in your IoT solution by setting up alerts by email, Slack, Teams, and other communication channels for communication or notifications
- Track and monitor events using other Microsoft SaaS services like Azure
- Integrate B2B features that work with other services like Azure Service Bus, Biztalk Server, Azure Functions, Azure API Management, etc.

Logic apps, as shown in **Figure 10-3**, integrate well with Microsoft Entra ID and APIM to route events and data to backend systems, which can be other resources like an app service, an Azure Function app, or supported SaaS services like Salesforce and Dropbox. Developers can work through APIM's developer portal. The API gateway can be integrated with Microsoft Entra ID.

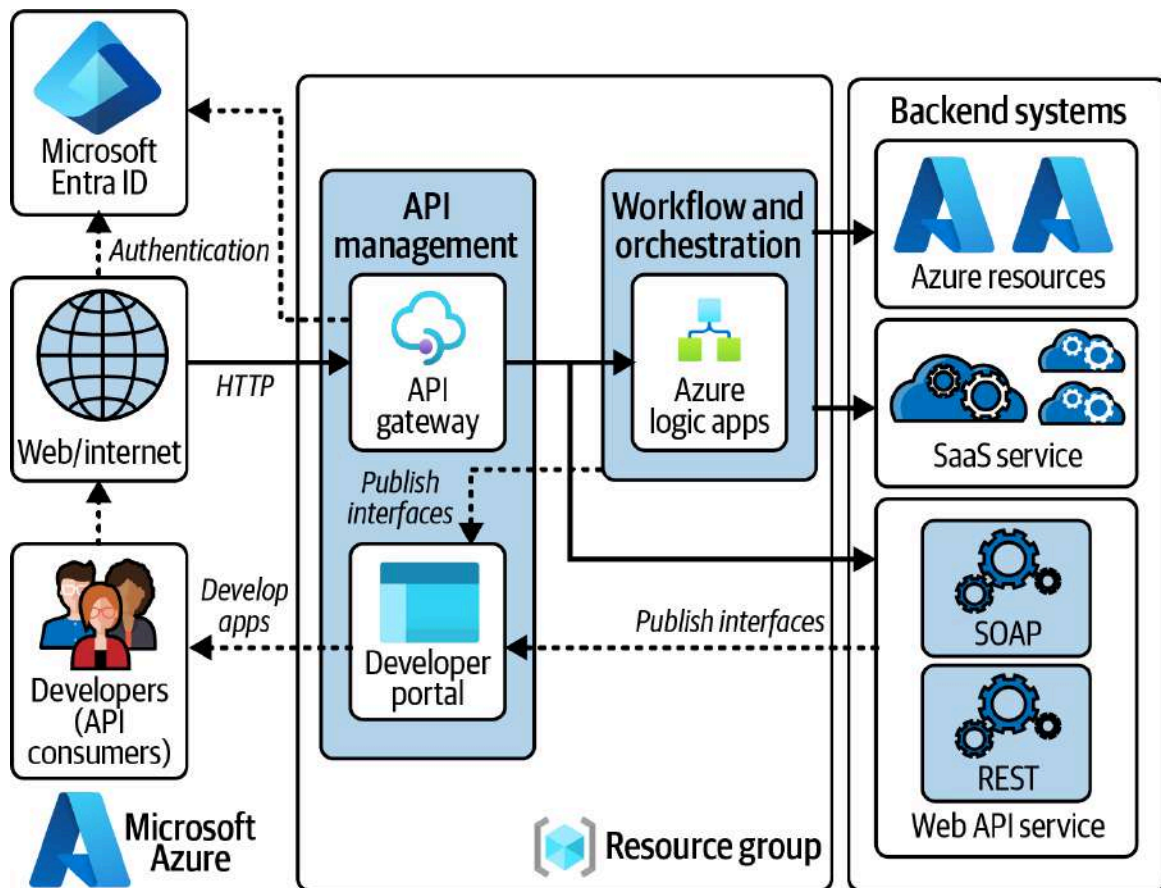


Figure 10-3. Enterprise integration in Microsoft Azure using Azure Logic Apps with other Azure services

Pricing Tiers of Azure Logic Apps

It is important to choose an appropriate Azure Logic Apps pricing tier by taking into account factors such as cost efficiency, scalability, integration capabilities, security, and compliance. Azure has two pricing tiers for creating Azure Logic Apps depending on your use case:

Consumption (multi-tenant)

This tier allows users to pay for what is being executed or used. It supports hosting **multi-tenant** and Integration Service Environmentss (ISE).

Standard (single tenant)

This tier supports single tenants, such as an App Service Environment (ASE).

By selecting the appropriate pricing tier, you pay only for the functionality and connections that you need, thus saving money.

WARNING

In **August 2024**, the ISE will retire because of its dependency on classic Azure Cloud Services. If you are currently implementing ISE with Azure Logic Apps in existing applications, please read this recommended **migration guide** by Microsoft.

The **Azure Pricing Calculator** is a great cost estimation and management tool for implementing Azure Logic Apps. This tool is also helpful when you are considering switching tiers, plans, etc., as business requirements demand. Microsoft also has a good **guide** for planning your costs for Logic Apps.

Azure Logic Apps Components

Azure Logic Apps has four key components: workflows, triggers, actions, and built-in connectors. These components allow for integration with various systems and applications, automation of business processes, visual depiction of workflows, and a low-code strategy for integration.

Workflow

A workflow is a sequence of steps describing the flow of a business process, task, or workload you want to implement. Workflows are visual representations of the automated business processes in Logic Apps. Workflows are essential because they depict the automation process clearly, making it easy for users to understand, alter, and sustain their organization's operations.

Trigger

Every workflow begins with a single trigger when using a logic app. A trigger is activated when a specific condition is satisfied, such as when a particular event occurs or data satisfies specific requirements. Numerous triggers have scheduling features that allow you to control the frequency with which your workflow is executed. Following the activation of the trigger a series of steps begin running and executing. These actions either handle the process, convert the data moving through the workflow, or move the workflow to the subsequent stage. The **Logic Apps workflow designer**, accessed through the Azure Portal, Visual Studio Code, or Visual Studio, allows for the graphical creation of workflows. The fundamental specification of each workflow is written in JavaScript Object Notation (JSON), and this definition is included with each workflow. Altering this JSON specification allows you to edit workflows if desired. Support for Azure PowerShell and Azure CLI commands is provided for operations relating to creating and managing Azure Logic Apps.

Action

An action is a task or job that needs to be executed during an Azure Logic App workflow, for example, sending an email through built-in connectors like Outlook or Office 365. Using this action step in Azure Logic Apps, you can set your logical condition, like the specific recipients to send alerts or notifications to, etc.

Built-in connectors

The built-in connectors that run natively in the Azure Logic Apps allow users to control the structure and manage the logic of the workflow. Logic Apps enable developers to

optimize business processes and incorporate services and systems by creating workflows. The built-in connectors allow developers to connect and integrate with various systems and services swiftly, which is one of the primary advantages of this Azure service. Such connectors are used with standard services such as Salesforce, Dropbox, and Twitter. The low-code/no-code features enable users to work efficiently without creating backend code and APIs. They provide a straightforward method for developers to access data and initiate actions within these services.

Advantages of using built-in connectors in Azure Logic Apps include:

- Developers are not required to invest time developing custom code or APIs to connect with different applications, saving time and effort during development.
- Built-in connectors include pre-built actions and triggers, making it simple for developers to integrate with different services. The integrated connectors are available for various services, allowing developers to construct flexible workflows.
- These connectors are also designed with support for security, providing assurance that the data is safely transmitted and stored.
- Built-in connectors are an integral component of Logic Apps and make it simpler for developers to integrate with different services, automate workflows, and create effective business processes

Some built-in Logic Apps connectors are not explicitly associated with any Azure or Microsoft service, and customization is flexible based on the use case. It is worth checking out the different built-in connectors available per pricing tier.

In the next section, you will learn about Service Bus, another excellent integration service in Azure that works well with API Management, Logic Apps, and other Azure services.

Azure Service Bus: Cloud Messaging Broker Service

One of the common reasons enterprises and organizations consider using cloud computing is for data integration between applications. Cloud computing enables integration through enterprise messaging services, which helps implement messaging to transfer data between different types of applications, APIs, or web services. A messaging service uses a message to transfer data between applications, regardless of whether they are hosted in the cloud, on premises, or hybrid. The data transmitted in a messaging service broker in the cloud can be in different formats such as plain text, XML, or JSON.

Service Bus is a cloud enterprise messaging service in Azure that serves as an intermediary broker applications and services can use to transfer data or information from one application to another. It can solve complex messaging problems that many developers previously had to solve programmatically regarding passing data, messages, or information between distributed enterprise applications.

Data integration between decoupled applications needs messaging that is reliable and secure. Service Bus guarantees secured, reliable, and scalable message delivery between applications through Azure's platform. It is possible to send or receive asynchronous messages or data from one to many systems using **queues and topics** in the Azure Service Bus service.

In addition, you can use one-to-many relationships for messaging delivery using topics and subscriptions. These will be compared in the coming sections.

Common uses of Azure Service Bus include:

- Serves as an intermediary broker that applications and services can communicate with to transfer data or information from one application to another. It can also be integrated with other resources within Azure, like serverless integration with web applications, IoT solutions, Event Grid, Azure Logic Apps, etc.

- Delivers messages for multiple applications. It supports several parallel competing consumers, meaning multiple applications can read from a queue. A topic can also be used for the publish-subscriber pattern, which is discussed later in this book.
- Provides reliable messaging for decoupled and distributed systems. Some applications are designed to be decoupled and distributed for reasons such as improving flexibility, scalability, and reliability. Azure Service Bus provides reliable messaging capabilities between multiple distributed or decoupled systems.
- Acts as a message broker in the cloud capable of handling transactions (operations grouped to be executed together) with atomicity. It also guarantees the integrity of message storages in its internal operations; for example, it supports a dead-letter queue (DLQ) in case of message failure. It is essential to note that transactions are supported only in the Premium and Standard pricing tiers.
- Supports one-to-many communication. Many business processes require the implementation of one-to-many communication within a publisher-subscriber 1:n relationship. Azure Service Bus topics and subscriptions support this scenario.

Service Bus has other advanced features, including auto-forwarding, which allows you to forward or chain messages between topics or queues in the same namespace, duplicate detection, pre-fetch, message deferral, etc.

Service Bus supports common programming languages for cloud development. Developers or cloud engineers can design, develop, and integrate systems using C#, Visual Basic, F#, Java, Go, C/C++, PHP, Ruby, and Java Messaging Service (JMS).

TIP

Azure Service Bus Premium Tier supports **Java Message Service (JMS) 2.0**, a standard service used in accessing middleware services for messaging in Java. You can integrate and communicate with Azure Service Bus from your Java applications using the Advanced Message Queueing Protocol (AMQP 1.0).

You can fully integrate the Azure Service Bus with other Azure services, such as Azure Event Grid, to respond to any events coming through Azure Event Grid and Azure Logic Apps. Service Bus also supports serverless development. For example, you can use it with Azure Functions using the Azure Service Bus bindings and its triggers for events happening in a specific resource or application.

Combining Azure Service Bus and Azure Functions allows you to develop a robust messaging and event-based architecture that enables real-time communication and processing between multiple apps and services.

Following are examples of how these can be combined and integrated:

Process messages between applications using queues

Azure Functions can be used to ingest messages from a Service Bus queue, process them, and perform necessary actions, such as modifying a table in a database, delivering an email, or triggering another Azure Function.

Trigger an Azure Function based on a queue message

Azure Service Bus can send queue messages to an Azure Function using the Service Bus trigger of Azure Functions. You can process the message and execute the necessary duties through this serverless integration. This is useful when you need to initiate an action based on a specific event, such as the arrival of a new customer order or the completion of a lengthy task.

Implementation of publish-subscribe pattern

Service Bus gives developers a publish-subscribe technique that can be used to facilitate immediate communication between services and applications. Subscribe to topics and handle messages as they arrive using Azure Functions.

Combining the features of Azure Service Bus and Azure Functions, you can develop a highly scalable, event-driven architecture that enables real-time communication and processing between cloud-based services and applications.

You can also add Service Bus in your low-code or no-code automated workflows using Azure Logic Apps to manage your messages or data. You can use Azure Service Bus with a Power Platform's Dataverse in Dynamics 365 trigger to create a decoupled architecture where events in Dataverse trigger actions in other services and applications through Service Bus and Power Platform. Integration with [Azure Stream Analytics](#) for queues and topics for processing outputs is also supported.

Depending on the use case, you can use Azure Service to transfer data or information immediately, on a schedule, or with delay, using enterprise message broker systems or services. If you are using .NET, you are most likely already using Azure Service Bus; however, Apache ActiveMQ is another type of message broker commonly used to develop solutions using Java for enterprise applications.

Azure Service Bus and Apache ActiveMQ are message brokers that work with message providers like JMS to send and receive messages between client applications. They have similar capabilities, like queues and publish-subscribe semantics. That said, they have some differences too worth checking out and considering before implementation; the details of the technical differences are beyond the scope of this book.

Microsoft's [guide](#) for migrating JMS applications from Apache ActiveMQ to Service Bus is a good resource to learn more.

Azure Service Bus Components

The Service Bus delivers a reliable and expandable messaging infrastructure that can be used to construct distributed apps and services. It provides secure and private communication between applications, enables several kinds of messaging patterns, and interfaces with other services offered by Azure to allow real-time information processing and analysis.

Following are the components that comprise the core of Service Bus's messaging capabilities: namespaces, queues, topics, and subscriptions.

Namespaces

An Azure Service Bus namespace is required before creating a queue, topics, or subscriptions for message processing or delivery in the Azure Service Bus service. A namespace is like a hub or the logical scoping container of your Azure Service Bus messaging services like the topics, queues, etc.

Figure 10-4 shows how data flows between the namespace and distributed systems. The advantage of the Service Bus namespace is that you can gather all your Azure Service Bus queues, issues, etc., for typical distributed applications in one place.

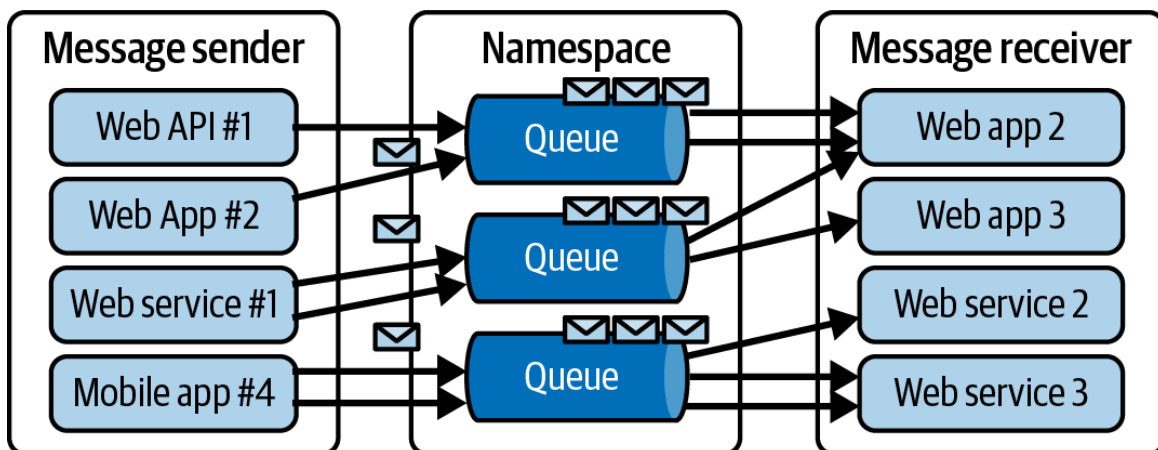


Figure 10-4. Azure Service Bus queues (multi-tier apps)

Queues

By design, distributed applications are not intended to communicate directly with only one application; thus, these applications can communicate and be integrated even if not in the same hosting or instance. A messaging broker, such as the Azure Service Bus, sends messages or data. When a distributed application needs to send a message to another application, it must use queue storage. In Azure Service Bus, a queue acts as the logical container of messages an application sends that can be stored, retrieved, processed, or retrieved from another application or web service.

An Azure Service Bus queue uses the algorithm concept of **FIFO** (first in, first out). When an application's message arrives at the queue, it is held in this cloud messaging queue storage until it is time to be retrieved in the same order that it arrived. This messaging service provides advanced features that help you manage the messages between distributed applications depending on each unique use case.

Figure 10-4 illustrates how Service Bus works in different multi-tier applications. Distributed applications, like a web app, mobile app, or web service, can pass and send data to another application using the Azure Service Bus queue service.

Topics and subscriptions

The Azure Service Bus topics, like queues, act as cloud messaging brokers. The difference is that the subject uses one-way communication, meaning it uses the subscription to deliver messages to its consumers or receivers, also called subscribers.

Publisher-subscriber is one of the cloud messaging architectural design patterns for distributed applications in the cloud. This design pattern is supported in Azure Service Bus with topics and subscriptions, where users can simultaneously send messages or events to interested consumers.

Topic and subscriptions is a feature that developers can use for establishing a publish-subscribe messaging pattern. This capability enables an

asynchronous, decoupled communication model between multiple components or applications.

Let us review how each works and operates:

- A *topic* is a logical entity representing a message stream. For example, when a message from an application is published to a Service Bus topic, it is sent to all associated subscriptions and subscribers.
- A *subscription* is a named view on a topic that specifies a subset of the topic's communications. Subscriptions allow applications and components to filter and receive only the messages that are relevant to them.

Using topics and subscriptions, developers can construct a messaging system in which publishers and subscribers are unaware of one another. A publisher can send messages to a topic and then to all interested subscriptions. Subscribers can filter messages based on various criteria, such as message properties or content, and receive only the relevant notifications.

This feature is useful when multiple components of an application or multiple applications need to communicate with one another but are not tightly coupled or dependent on one another. Examples of practical uses in which topics and subscriptions may be utilized include:

- A news publishing platform in which different news categories are published on various topics, and subscribers can choose to receive only the news that interests them.
- A stock trading application in which users subscribe to the equities they are interested in and receive updates only for those stocks.
- An IoT application in which various devices publish data on distinct topics, and various applications or components can subscribe to these topics to receive the data that is pertinent to them.

Overall, topics and subscriptions in Service Bus provide developers with a robust messaging functionality that promotes a decoupled, **asynchronous communication model** between components of an application or between applications.

Choosing the Right Azure Cloud Messaging Implementation

Azure Service Bus is beneficial in brokered messaging for decoupled applications, enabling better data transfer, communication integration, and other helpful cloud messaging features. Choosing the right cloud messaging implementations can take time and effort. For example, when choosing between an Azure Service Bus queue or Azure Storage, you should assess how your use case fits each service's features.

In most cases, you would choose Azure Service Bus over Azure Storage Queue if you need a FIFO guarantee of your queue messages. You also would use Azure Service Bus to group your messages into transactions or if the use case requires you to consume a batch of multiple messages for your distributed applications.

On the other hand, Azure Storage is preferred over Azure Service Bus queues if, for example, your use case needs a standard queue without any required additional features. Consider Azure Storage Queue over the Service Bus queues if your queue message exceeds 80 GB.

Service Bus queues can handle messages only between 64 KB to 265 KB if you use the Standard pricing tier; if you have the Premium tier, you can send up to 100 MB. Please note that these message capacity per service tiers may change.

In the next section, you will learn about Azure Web PubSub, another cloud messaging service in Azure that is ideal for building web apps with real-time features.

Azure Web PubSub

Azure Web PubSub is a managed service within Azure that enables developers to create scalable real-time messaging applications. Web PubSub enables developers to transmit and receive messages in real time, allowing them to develop applications that require instant updates, such as chat applications and live dashboards.

Web PubSub provides numerous advantages, such as simple integration with other Azure services, built-in security features, and the capacity to manage millions of connections and messages. Web PubSub is typically used for chat applications, real-time monitoring, and live transmission.

There are three pricing tiers for Web PubSub: free, standard, and premium (public preview).

Free

The free tier supports up to 20 connections, 20,000 messages per day, Max Units of 1, and an SLA of 99.9%, which makes it ideal for testing and development.

Standard

The standard tier offers increased performance and scalability, with up to 1,000 connections, unlimited messages per day (the first 1 million messages are complimentary), and Max Units of 100 with an SLA of 99.9%. It also incorporates message broadcasting and message ordering capabilities. The standard plan is billed according to the number of connections, messages, and data transferred.

Premium

The premium tier offers the same features as the standard tier with additional features such as enhanced message reliability, fully-managed autoscaling, customized domain name, and availability zone support. The premium tier is

meant for applications that require the most significant degree of reliability and performance for enterprise production workloads.

In the previous section, you learned about Azure Service Bus, a reliable messaging service for sending and receiving messages in the cloud between distributed applications. These messages can then be delayed or processed based on preferences and system design.

Web PubSub, on the other hand, is a PaaS service in Azure that enable its users to subscribe or send-receive messages between client and web services in real time using **WebSockets**. As a persistent web communication protocol, a WebSocket uses a TCP connection between a web client and a web server for bidirectional real-time communication.

Figure 10-5 shows that WebSockets allows live contact between the web client and server. As illustrated in the figure, Azure Web PubSub uses the following sequence:

1. The web client sends an initial HTTP request to the web server.
2. The web server confirms and accepts the request from the client.
3. When a client-server handshake is created, the client and the server can communicate and exchange messages in real time.

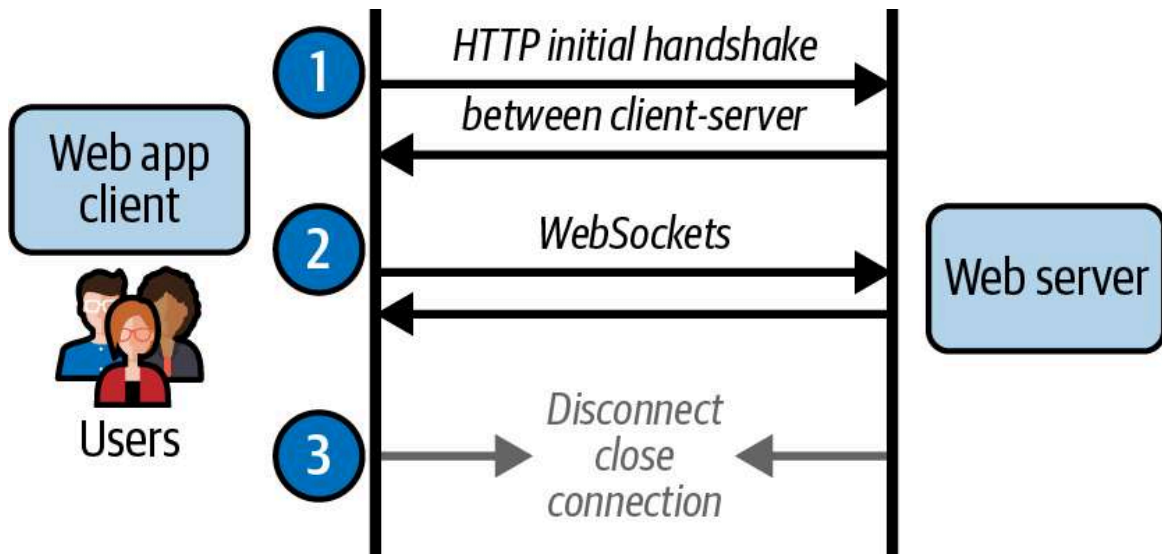


Figure 10-5. Azure Web PubSub web communication via WebSockets

WebSockets are preferred over HTTP connections because of their speed. WebSockets are bidirectional, while HTTP connections are unidirectional. HTTP communication takes longer because data must be requested before the web server can send data or messages back to the client.

Azure Web PubSub uses serverless and native WebSockets, enabling developers to fully manage it on Azure's cloud platform and develop it with their applications. It supports publish-subscribe messaging and use cases for web applications and mobile apps requiring real-time data transfers or communication.

The Architecture Pattern Used in Azure Web PubSub

Like the pattern used in the topics and subscriptions of Azure Service Bus, the publisher-subscriber pattern (also called pub/sub messaging) is used in Azure Web PubSub. It allows us to build modern applications that relay or send messages to many subscribers asynchronously, as shown in **Figure 10-6**, where the publisher sends a message to an input channel. The input channel sends the message out to the interested subscribers through a message broker and an output channel.

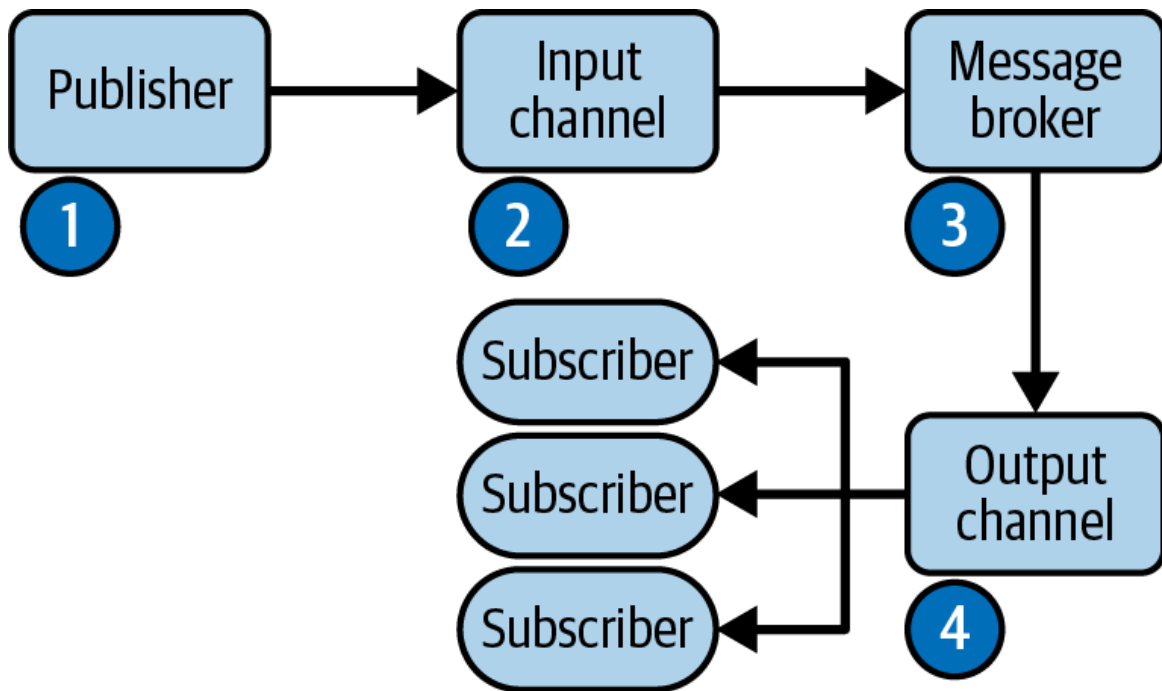


Figure 10-6. Publisher-subscriber pattern sends/receives messages from one application to its multiple interested subscribers

The publisher-subscriber steps are as follows based on **Figure 10-6**:

1. The application that acts as a sender (publisher) sends a packet of data events in the form of messages. These messages sent will be transmitted to the input channel.
2. The input channel communicates with a message broker.
3. The message broker then copies or replicates these messages to an output channel.
4. The output channel then sends the message to its consumers (subscribers).

Using the publisher-subscriber pattern, distributed and cloud-based applications can send messages from/to different systems components in async. This pattern helps avoid interruptions. It can assist in blocking publishers from waiting for any response from any subscriber since they are decoupled and isolated from the subscribers or receivers.

The publisher-subscriber pattern also provides advantages such as increased reliability through support for asynchronous messaging and the ability to handle failures efficiently through the use of message brokers in the cloud. Consider using this pattern to build applications that send information to multiple consumers. This pattern is ideal for consistent data processing for your applications.

Benefits of Azure Web PubSub

Web PubSub lets you quickly develop real-time messaging web apps by utilizing WebSockets. This real-time functionality enables publishing content updates between the server and connected clients, like an app for mobile devices or a web application with a single page.

Uses for Azure Web PubSub include:

- Real-time messaging between clients and web servers
- Build applications with live data dashboards
- Reliable and high-frequency updates of data, content, feeds, and reports, e.g., live auctions, gaming, voting polls, etc.
- Multi-platform real-time collaboration, e.g., live pair programming workspace
- Live chat and real-time communication, e.g., online customer service, web assistant, etc.
- IoT real-time monitoring
- Send real-time alerts, notifications, collaboration tools, etc.

In addition to these use cases, Azure Web PubSub also allows client-to-web server connections for large-scale applications with support for scalability, high availability, the ability to create many instances, and client connections.

Developing for Azure Web PubSub is flexible and easy because of its broad support for different types of clients. For example, it supports web client apps for web, desktop, mobile, IoT, servers, and game consoles.

Fundamentals of Azure Web PubSub

To develop with Azure Web PubSub, you must first understand some essential concepts, terms, and components.

Server

The Web PubSub service's primary purpose is to handle the connections, events, messages, etc., communicated and transmitted from or to the web client application. The server in the Azure Web PubSub service can play the role of the server-side event listener and event handler.

Message

A message is data in text format, JSON, or binary. When the client has established the connection to the server using the WebSocket connection, it can send messages to another application. The maximum message size that can be sent is 1 MB.

Hub

A hub is a logical concept for grouping places in your client connections based on purpose. For example, you may create a hub for client connections related to chat communications, notifications, or alerts. Each hub can be associated with an Azure Web PubSub service. You can associate different applications in unique hub names within the same Azure Web PubSub service.

Group

Aside from the hub in Azure Web PubSub, you can also create a group. The client can join or leave the group anytime. In Azure Web PubSub, a group session can have multiple clients, and a client can join or connect to different group sessions simultaneously. In this case, if a message or notice is sent to a group, it will be received by clients connected to that group session.

Client events

Any event initiated by the client is considered a client event. Examples of client events include a connection through WebSocket, sending or receiving messages, connection states, etc.

Event handlers

An event handler in Azure Web PubSub handles the logic of incoming events happening in the client. Event handlers manage the client events that should be filtered for further processing. To integrate the event handlers with client events, it is necessary to register and set up the event handlers through Azure CLI or the Azure Portal beforehand.

Event listeners

Event listeners wait and listen to incoming client events. The Web PubSub service listeners cannot interrupt communication between the client, the service, and the server.

Users

Azure Web PubSub connections are associated with a user. Each user can connect to multiple devices, browsers, or sessions.

Now that you have a handle on the essential concepts, you can learn about the typical workflow of Azure Web PubSub.

Typical Azure Web PubSub Workflow

A typical connection workflow for Azure Web PubSub, as shown in **Figure 10-7**, starts with the following sequence:

1. Client initiates the request to connect to the service.
2. The service invokes the server using the Cloud Events protocol, an event handler over HTTP.
3. The server gives a response to the service through the REST API.
4. The service sends the message response from the server to the client.

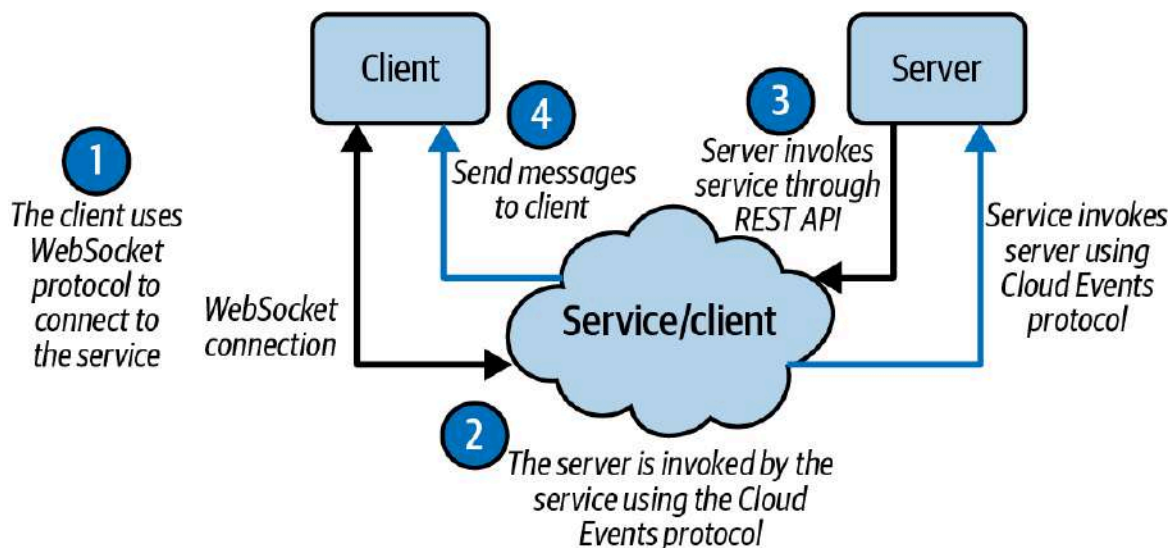


Figure 10-7. Azure Web PubSub workflow diagram

Azure Web PubSub serves as a broker on WebSockets and REST that enables integration of most programming SDKs and programming languages. Initially, you need to create the Azure Web PubSub service in Azure. You can make it directly on the Azure Portal, the Azure CLI, or deploy it as infrastructure as code using a Bicep template.

In the later chapters of this book, you will learn more about infrastructure as code technologies such as Azure Bicep and other cloud command-line tools

for Azure.

Azure Event Grid

If you want to create and build applications that support an event-based Pub-sub pattern, then Azure Event Grid is an excellent service to consider. It allows you to develop applications reactive to changes or events. You will typically use Azure Event Grid if you want to create actions when a specific event occurs, such as sending an email if significant events are happening with an Azure VM.

Event Grid integrates well as the event communication layer between its event resources and the different types of event handlers, such as the workflows, message brokers, Azure Automation, and other supported handlers. This event-tracking service can be integrated well with other services in Azure or even third-party or on-premises services by routing events to its distributed subscribers or registered receivers.

You can integrate applications using events with Azure Event Grid, a highly elastic, serverless computing event broker. Event Grid delivers events to subscriber destinations such as apps, Azure services, and any endpoint with network access.

The following are benefits of using Azure Event Grid for cloud integration development:

- Event-driven notification and messaging delivery that supports HTTP-based technologies within Azure and also offers support for external APIs or integration
- Reactive programming and development that helps you build better, more reliable applications, capitalizing on guaranteed event delivery and the cloud's high availability.
- Simplicity and flexibility for event-based applications. Event Grid can be used as a managed service for managing the distribution of events

from any source to any destination, simplifying your event-based applications.

- Minimize latency by eliminating polling because Event Grid decouples applications using a pub-sub pattern and simple HTTP-based event delivery, enabling the development of scalable serverless applications, microservices, and distributed systems

Building with Azure Event Grid for integrating applications to react to certain events can benefit organizations that want to extend their business processes and implementations through custom event handles, customized events, etc.

Learn By Doing (Try It!)

The following are recommended tutorials and guides from Microsoft's official documentation.

- [Tutorial: Import and publish your first API](#)
- [Tutorial: Publish and subscribe messages using WebSocket API and Azure Web PubSub service SDK](#)
- [Tutorial: Create a serverless notification app with Azure Functions and Azure Web PubSub Service](#)
- [Tutorial: Create a chat app with Azure Web PubSub service](#)
- [Tutorial: Monitor virtual machine changes using Azure Event Grid and Azure Logic Apps](#)

Summary

In this chapter, you learned how cloud integration technologies and services make it easier to collaborate and synergize different business processes and systems flexibly. Applications that adopt cloud integration tools benefit from improved scalability, reliability, cost-efficiency, and business agility

by getting more flexible opportunities to connect, transform, and integrate their data and other business processes into other applications.

You learned about some of the cloud integration services and tools available in Azure, including Azure API Management, which offers comprehensive API management capabilities for consumers, developers, and API providers, and Azure Logic Apps, a low-code/no-code integration platform that enables users to create stateful serverless workflows with ease.

Additionally, the chapter covered Azure messaging services such as Azure Service Bus and Azure Web PubSub, which offer different ways of sending messages to distributed applications, web services, APIs, or systems.

Finally, the chapter highlighted the benefits of using Azure Event Grid for event management with applications, allowing for better control of and reactivity to changes and events, whether on Azure or on premises.

Check Your Knowledge

1. What are the benefits of cloud integrations to applications on premises, hybrid, or in the cloud?
2. How can Azure API Management help with the development of applications with backends that are suitable for a microservices architecture?
3. Based on what you have learned about Azure Service Bus, what other helpful use cases exist for brokered messaging technology?
4. What Azure integration service would you consider using if you were to build a collaboration dashboard where users can join, connect, and collaborate in real time?
5. When would you consider Azure Event Grid versus Azure Event Hub?

Answers to these questions are in the [Appendix](#).

Recommended Learning Resources

Andersson, Jonah. “Azure Service Bus – A Brief Technical Overview.”

Jonah Andersson Tech, https://oreil.ly/q17_t.

“API Management Documentation.” Microsoft Learn, <https://oreil.ly/Hffil>.

Azure API Design. Microsoft Azure ebook, 2019, <https://oreil.ly/sEb3C>.

“Azure Service Bus Messaging Documentation.” Microsoft Learn,

https://oreil.ly/s_9U5.

Azure Web PubSub Demos, Microsoft Azure, <https://oreil.ly/T0rgN>.

Malvik. Sven. Mastering Azure API Management. Berkeley, CA: Apress, 2022.

“Quickstart: Create a Web PubSub Instance with the Azure CLI.” Microsoft Learn, January 12, 2023, <https://oreil.ly/sL49U>.

Russinovich, Mark. “Microservices: An Application Revolution Powered by the Cloud.” Microsoft Azure blog, March 17, 2016, https://oreil.ly/nl9_X.

“The Samples for Azure Web PubSub.” GitHub repository,

<https://oreil.ly/Y5Bcv>.

Chapter 11. Cloud Infrastructure, DevOps, and Monitoring in Azure

We should never doubt, diminish, or dismiss the value and importance of people—their ability to think will always eclipse the capability of your tools.

— **Andrew Urwin**, Microsoft Azure MVP and Director of Platform and DX at Clue Software

*Written by the author with contributions from Andrew Urwin, Engineering Leader, DevOps Evangelist, and Microsoft Azure MVP; and Freek Berson, author of **Getting Started with Bicep: Infrastructure as Code on Azure** and Microsoft MVP for Azure and Enterprise Mobility*

Introduction

In this chapter, you will learn more about how Azure can be used with its different tools and services for IT operations, cloud development, collaboration, automation, and more. After you have learned about Azure's various cloud integration tools and services, you will learn more about cloud infrastructure automation tools, modern DevOps practices, building with cloud-native infrastructure, and monitoring in Azure.

Cloud-Native Infrastructure

IT infrastructure refers to all the components needed to operate and manage IT environments. It can include data centers, web servers, databases, applications, or any physical resources needed to keep IT systems running.

Moreover, an organization's IT infrastructure can be hosted on premises, natively in the cloud, or both.

When you have an on-premises infrastructure, you must deal with the hassles of the up-front costs (CapEx) of physical IT infrastructure. When your on-premises infrastructure grows, the costs for upgrading it may also increase as it expands.

On the other hand, with cloud-native infrastructure, everything you need to run your IT infrastructure is operated through tools and services designed for using the cloud. The benefits are the same as those described for cloud computing in **Chapter 1**, which explained the differences and benefits of hosting applications on the cloud and the cost advantages.

By deploying and hosting applications or systems to a public cloud provider such as Microsoft Azure, organizations can avoid the costs associated with managing physical components while gaining the cloud computing benefits of scalability, flexibility, reliability, disaster recovery options, speed, and so on, at global scale.

Cloud computing allows organizations to focus on delivering business value quickly while enabling teams to work and collaborate effectively with everyday IT operations assisted by cloud computing.

Developers don't concern themselves with where organizations host their systems or applications (cloud, on premises, or hybrid); typically, they just want to do what they enjoy: code and provide solutions to customers, solve business problems efficiently, and maximize their productivity.

Developers prefer to reduce time spent on manual and repetitive tasks on the infrastructure. For developers who are still working with applications hosted on an on-premises infrastructure, it is possible that they still have to deal with the manual way of working with traditional application development.

Dealing with the manual process of deploying code that was finished and developed without the automated integration of CI/CD from source code

version control is very tedious for a developer. In addition, manual tasks are prone to errors.

Developing and building systems for the cloud usually gives leverage to developers, organizations, and diversity teams within that organization. Cloud development and operations—from development to delivery—provide many opportunities for learning while building modern systems that provide value to the business and its end users.

An IT infrastructure (on premises or in the cloud) comprises many technical components. Such components include resources for computing, networking, databases, servers, storage, applications, etc. Each technical detail is essential in building the entire infrastructure of the application or system, whether for small, medium, or large enterprises.

Organizations, companies, and IT teams can benefit from a well-designed and well-built IT infrastructure, especially for hybrid or cloud systems. The benefits include not only the modern capabilities that cloud computing providers such as Azure provide, but also improved end-user satisfaction, improved business processes, opportunities for better collaboration, and opportunities to market globally quickly.

TIP

Check out Microsoft's [ten recommended design principles for Azure applications](#) that will help design an infrastructure for your applications so they are more manageable, scalable, and resilient.

While a well-architected IT infrastructure on Azure provides these benefits, designing, structuring, and building infrastructure requires proper planning. Like any project, if poorly designed, planned, and built, it likely will not be durable or sustainable.

At a high level, developing cloud-native infrastructure involves fundamental practices, concepts, or methods such as:

Cloud-native for microservices architecture

In certain use cases an architecture that implements microservices can use the benefits and solutions of a cloud platform such as Azure. Instead of monolithic applications, cloud-native infrastructure emphasizes breaking up applications into more minor, loosely coupled services that can be independently developed, deployed, and scaled.

Deploying and containerizing applications to the cloud

One benefit of maximizing Azure is the capability of implementing cloud-native solutions. An example of practical, cloud-native implementation is deploying applications into containers. Cloud-native infrastructure relies heavily on containerization technology, such as Docker, to package and deploy microservices. The advantages of using containers are consistency of environment configuration and processes in different cloud environments, giving more flexibility to varying teams regarding upgrades, improvements, and managing dependencies.

Cloud-native for orchestration and scalability

To manage containers at scale, cloud-native infrastructure uses orchestration tools for containers, such as Kubernetes in Azure, to automate the deployment, scaling, and administration of containerized applications.

Infrastructure automation

Automating infrastructure, such as implementing infrastructure as code (IaC), is another benefit of cloud-native development. IaC enables consistency, automation, and better management of resources on the cloud. Some IaC providers and tools support multi-cloud deployments that can be helpful to projects that want to prevent cloud vendor

lock-in. Automating resources in the cloud makes repeating and deploying these environments easier than doing so manually. Terraform is an IaC provider that provides multi-cloud platform deployments; however, ARM/Bicep is dedicated to a single cloud platform such as Azure. Of course, there are reasons we have these options. Choose the best possible IaC that suits your use case and business requirements.

These cloud-native practices may change as they are developed, and they can be combined or implemented depending on the use case.

Modern Application Development and DevOps

The idea of **DevOps** was started in 2007 by Patrick Debois from Belgium. He was learning about IT from different perspectives and realized the frustration of IT projects that require switching between people working with IT operations and developers.

When Patrick met Andrew Shafer in Toronto in 2008 at the Agile Infrastructure Conference, they started discussing the drawbacks of Agile. They exchanged and brainstormed ideas on how they could solve the issues of *Dev* and *Ops*. In 2016, Patrick became one of the authors of *The DevOps Handbook*.

In DevOps, developers play a vital role in delivering software and applications, whether for hosting to the cloud or on premises. They also need to understand the business requirements and convert them into code logic, which can later be tested, deployed, and delivered to its end users for production and the real world. Therefore, developers and engineers working with infrastructure need a way to collaborate and effectively work to deliver software. This is where DevOps practices and processes come into play.

DevOps is a methodology that integrates and automates the work of software development and IT operations to streamline and accelerate the software development process. At the same time, it helps collaboration and communication between development and operations teams.

The term DevOps is a practice, and people in the IT industry have different opinions about it. Regardless, it is a practice that plays a vital role in any IT project and team collaboration. The terminology may change as technology advances, but we are developing and building these technologies. The most important thing is that teams find the best and most flexible way of collaborating to deliver value and purpose to the project.

In modern cloud development, the standard practices we encounter involve Agile application development, continuous integration (CI), continuous deployment (CD), infrastructure automation, and many other cloud-native tools. Whether you're a developer, DevOps engineer, or cloud engineer, you can work with any of these processes in combination together with other team members. The most important key in this collaborative process is being open to change and learning from each other.

The Core of DevOps and Its Function in Application Development

The development of cloud applications requires cross-disciplinary engineering practices. DevOps is a practice where developers, systems administrators, cloud administrators, IT managers, etc., collaborate on a project. Although the label "DevOps" is being used in many instances, it is not a title but a practice requiring a wide variety of technical and nontechnical knowledge.

The core of DevOps includes CI/CD, automation, and infrastructure management, especially in the cloud. Continuous testing and monitoring capabilities are also necessary. However, working with DevOps is more than just working on these things. There is a huge demand for experts with technical competencies to work with DevOps in many organizations today.¹

A trending topic is whether the word “DevOps” should be included in someone’s title. Perhaps the title “platform engineer” or “infrastructure engineer” should be used instead.² Right or wrong, it does not matter. What matters is that the teams are working together and are satisfied with their tasks within the project regardless of what they do.

In the next section, we take a closer look at these core processes of DevOps in developing applications for the cloud.

Continuous Integration, Deployment, Testing, and Monitoring

CI/CD processes start from development, where code changes are automatically built, tested, and deployed to production environments. Different robust and fully managed cloud-based CI/CD tools typically work well with Azure’s cloud platform. Cloud-based CI tools include GitHub, Azure DevOps, Jenkins, JetBrains, TeamCity, Octopus Deploy, and others. These tools make it easier for its users to configure, set up, and automate the DevOps CI/CD processes for developing, deploying, testing, and delivering software to their respective environments.

Continuous integration

Continuous integration (CI) means frequently integrating the code changes done by developers working in parallel and committing them into a shared code repository. CI is typically linked to continuous delivery (CD) or the deployment pipeline, discussed in the next session.

In a CI pipeline, the CI system automatically compiles the code, runs automated tests, and deploys the application to a testing environment whenever a developer pushes new code changes to the shared code repository. This allows developers to detect and fix integration problems and bugs in the early phase of the development process before they can cause expensive problems in production.

By enforcing CI in the cloud, developers can gain the benefits of scalability and flexibility of infrastructure on the cloud, which can automatically spin

up new testing environments and provide resources for testing and deployment. This allows developers to focus on writing high-quality code while the CI system builds, tests, and deploys code changes consistently and reliably.

Today's applications or systems involve different skill sets, people, practices, etc. Regardless of where it is hosted (cloud, hybrid, or on premises), application delivery requires other teams: engineers, developers, analysts, business, infrastructure, IT operations, etc. A developer builds the application based on the designed and planned architecture.

For example, team members must deliver their tasks in every planned sprint in a project using Agile or Scrum practices. Developers must plan the development tasks or part of the application's functionality and consistently deliver it for further testing, quality assurance, etc.

The benefit of CI is that it enables faster feedback to developers and different team members that will help them fix issues, bugs, or problems immediately. Another benefit of the CI process is that it helps improve code quality by detecting and identifying issues with integration with other systems or web APIs at the early stage before they get deployed.

Furthermore, using cloud-based tools and services for DevOps, developers can use on-demand computing resources to quickly spin up and tear down test environments, making more efficient use of resources.

Figure 11-1 shows how developers build the application components in their local development environments. Once they are done and have committed their final changes to the remote repository, the CI process enables the automatic deployment of their code to the development environment or server for the integration build. This will be processed in the next step, continuous deployment.

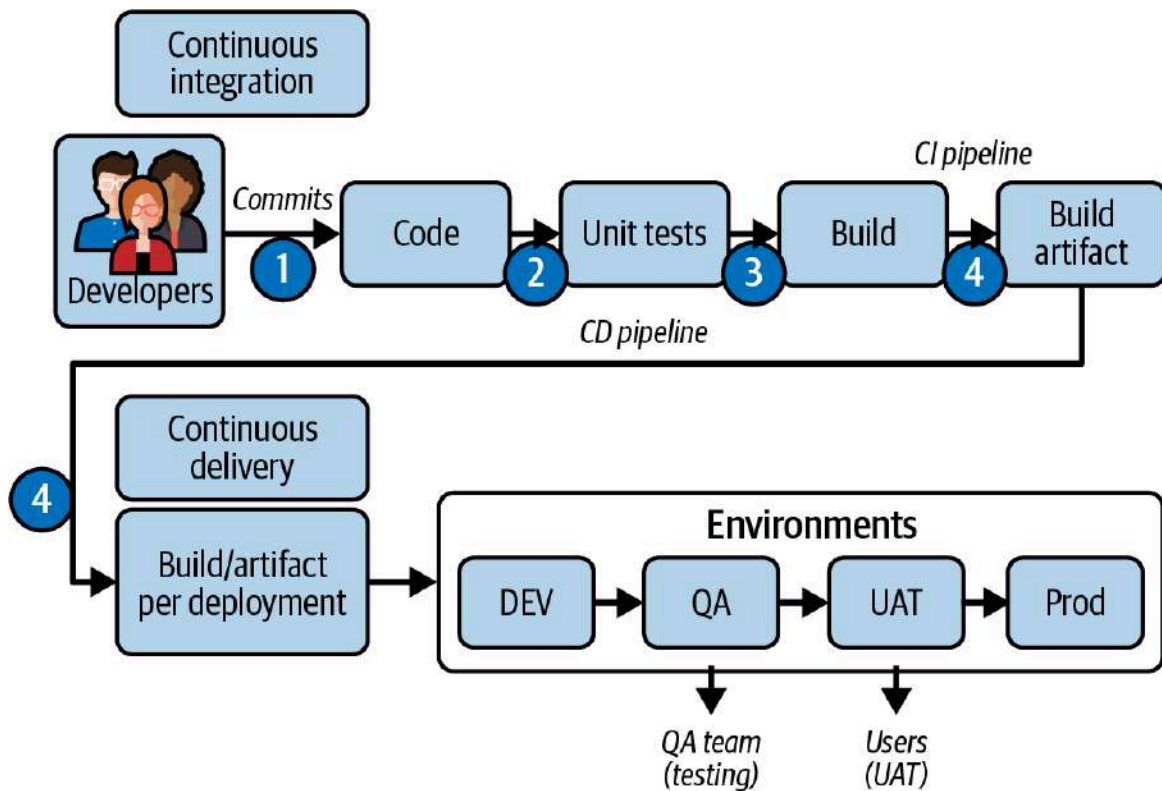


Figure 11-1. CI workflow in a typical IT project

A consistent cadence of integration is essential in any Agile IT project and for DevOps engineering practices. Continuous integration is a complex and vital part of the systems development lifecycle, or **SDLC**.

Implementing CI is a good tool for software development teams who want to increase collaboration and code quality to improve and accelerate development and release management.

Continuous deployment

Continuous deployment (CD), sometimes called continuous delivery, is a software development practice that automatically deploys every change developers make to a production environment. This approach automatically releases any code change that passes automated tests into production without manual intervention. The process of CD is typically implemented using automation tools and best practices related to deploying the developed systems or applications to their target environment.

Figure 11-1 illustrates how the software development workflow and steps relate to CD. As you can see in the figure, after the developers have pushed and committed the updates to the application, source code management is configured to build automatically. If you work with CD in Azure DevOps pipelines, the build is packaged as artifacts.

The deployment is handled by the release pipelines set up to deploy to different stages. These stages can be other web servers or environments in the cloud infrastructure, including Microsoft Azure.

Usually, the application is built and deployed automatically in the development environment. By doing this, developers can verify, right after they have committed their code changes, that what they made is working as it should before it gets deployed to the next stage of the environment, which is usually a QA or test environment for quality assurance testing.

The QA team's testing process uses known testing methods like integration, unit, regression, performance, etc. Usually, QA team members are responsible for testing the application in the QA or testing environment to deliver feedback on whether the new functionalities being developed are approved. If the testing goes well, the application is deployed to the next environment, such as staging, where the business users can test further. Staging is also known as the user acceptance testing (UAT) environment.

Not all IT projects have the privilege of dedicated QA teams that test the quality of the released functionality, and the testing processes may vary from one task to another. The most important consideration is ensuring the testing process is part of the application development cycle to ensure quality.

By implementing CD, developers can significantly reduce the time between writing code and getting it into the hands of users. This can help teams move faster and release updates more frequently while improving the quality of their code through regular testing and infrastructure automation, which we will learn more about in the next section.

Continuous testing

Testing can be manual or automated. Best practice is to use automated testing for performing the tasks that need to be tested.

As part of the CI/CD workflow, continuous testing integrated within the build are an essential practice; development pipelines are another important practice.

Different types of testing methods can be applied in different ways. This includes unit testing of the individual code units, such as methods and functions, and verifying if they behave as they should. Integration testing is also necessary to integrate APIs, web services, and systems that work together. Testing if the application meets the functional business requirements is called functional testing. There are also tests for security, performance, and much more. Testing is crucial because it helps detect issues and bugs early on to guarantee that the resulting product is of high quality and meets users' needs.

An essential part of the CI/CD pipeline involves using automated testing tools and practices to test code changes in real time or when needed. This helps identify problems early on before they become more costly and time-consuming.

Continuous testing involves using tools and frameworks to automate the testing process. These include testing frameworks like JUnit, Selenium, or TestNG and testing services like [Azure App Center](#) or BrowserStack. These tools assist testers and the quality assurance (QA) team to perform and run several tests in parallel. Most of these testing tools help its users detect, report, and notify issues that can verify the code's reliability, observability, quality, and performance.

Application testing is essential because it validates the application quality built and deployed to production. Developers can detect and fix issues quickly by integrating continuous testing into every development stage before deploying applications to Azure. When done correctly, it can assure that the final product delivered to users in production is of the highest possible quality.

Continuous monitoring

After the application has been deployed and delivered to production, the work does not stop. It needs to be improved and maintained all the time. Once the new application has been delivered to production, another phase begins: maintaining and monitoring it for your users.

As part of development and operations, continuous monitoring involves monitoring your applications' reliability, health, performance, and security regardless of whether they are fully hosted on Azure. It means you have to use the monitoring and performance tools of your cloud resources and its infrastructure to identify issues and errors that happened behind the scenes and take corrective action.

Achieving this involves automated monitoring tools and practices typically integrated into the CI/CD pipeline and monitoring tools integrated within the cloud platform where the applications, APIs, and web services are hosted.

TIP

Azure Monitor is a monitoring service within the platform. Microsoft Defender for Cloud is another security monitoring tool for security risks and threats to Azure resources.

Monitoring proactively is a good practice for keeping your systems robust, secure, durable, and reliable. These proactive monitoring practices ensure that an application consistently functions at its best and that any issues or problems are detected.

Overall, reliable systems should be durable and built to heal themselves. Any possible issues that automation can fix are ideal, and problems that require human interaction should be addressed as quickly as possible. Monitoring helps identify what needs improvement in the systems or applications we build for end users.

Azure DevOps

A wide range of DevOps tools and techniques can help developers build and deploy applications efficiently. Using standard tools where different teams can collaborate is a good place to synchronize and work effectively on a project, including the business team members and stakeholders.

Therefore, the cloud-based Azure DevOps suite is a good DevOps platform for different teams within the same IT project. It provides a comprehensive set of tools and services for developing, testing, and deploying cloud-based applications.

The Azure DevOps suite has different components and tools for everyone in the IT project working with Azure. As shown in **Figure 11-2**, the Azure DevOps Dashboard provides many features: Azure Boards, Azure Pipelines, Azure Repos, GitHub Advanced Security for Azure DevOps, Azure Test Plans, and Azure Artifacts.

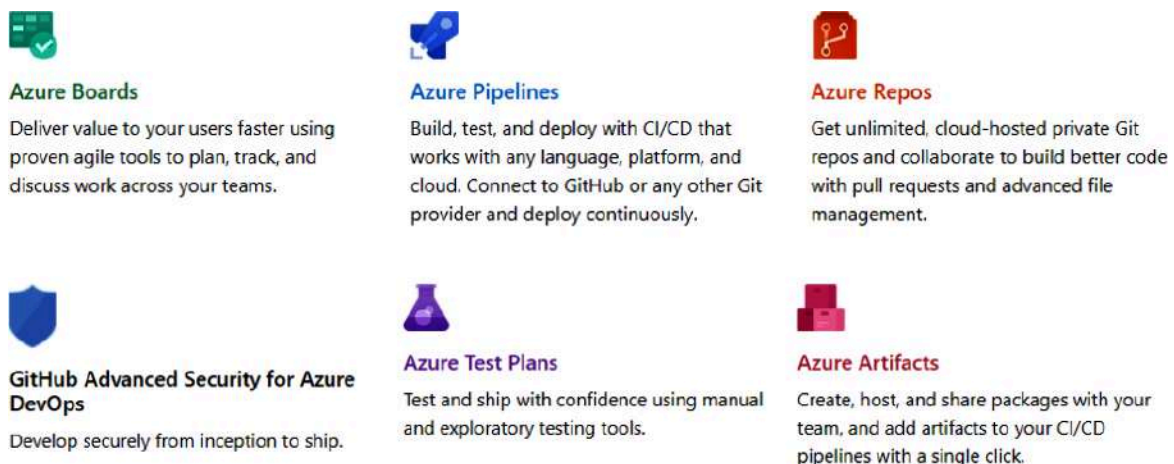


Figure 11-2. Azure DevOps suite's dashboard

Following is more information about each feature:

Azure Boards for Agile or Scrum IT project management

The Azure DevOps suite is an IT project management tool you can use to track tasks with your team. Azure Boards helps teams plan, track, and manage their software development projects. This tool is ideal for sprint planning,

task backlog management, and other features that will help your team stay organized. Azure Boards is integrated with different components, such as Azure Repos, to keep track of which commit is associated with specific tasks.

Azure Pipelines for CI/CD

Azure Pipelines supports automated CI/CD pipelines, which allow developers or DevOps engineers to develop, build, test, and deploy code changes to production automatically. It has extra support for different types of programming or source code projects, which means that it should be seamless to configure the build (CI) and deployment (CD) pipelines from your primary source code repositories within the Azure DevOps suite itself or if they are hosted in another source control systems.

Azure Repos for source code version control

Azure Repos is a powerful version control system that allows developers to manage their source code, repositories and environment configuration, track committed changes, collaborate with pull requests on code changes, and much more.

Azure Testing for managing tests

Azure Testing helps testers and the QA team in performing automated tests, tracking test results, taking action, and managing different test cases of the project. Using this tool within Azure DevOps can be beneficial in proving software quality and capturing any possible bugs, errors, and issues in the early stages of application development.

Azure DevOps built-in analytics and reporting

Azure DevOps suite has built-in analytics, auditing, and reporting tools, which allow teams to track key metrics such as code quality, test results, and deployment frequency. This helps teams identify improvement areas and make data-driven decisions about the software development process.

Overall, Azure DevOps is a powerful and flexible platform for cloud development, providing a wide range of tools and services to support Agile development and DevOps practices. Using Azure DevOps, development teams can streamline and automate their software development process, improve collaboration and communication between groups, and deliver high-quality software to users and stakeholders more quickly and reliably.

Azure DevOps Cloud-Based and On-Premises Solutions

There are two versions of Azure DevOps: DevOps Services and DevOps Server. The differences between DevOps Services and DevOps Server depend on the hosting type you need for your current infrastructure.

DevOps Services

DevOps Services is a cloud-based offering, meaning that Microsoft hosts all the development and deployment tools and services in the cloud. It provides a scalable and flexible platform for teams to build, test, and deploy applications and supports public and private cloud environments. With Azure DevOps Services, users can access all the tools they need through a web browser or a desktop client, and there is no need to maintain an infrastructure.

DevOps Server

DevOps Server is an on-premises offering, meaning users must install and maintain their infrastructure to host the platform. This can be a good choice for organizations that must keep their data and applications on premises for

regulatory or compliance reasons or for organizations that need complete control over their infrastructure.

DevOps Services delivers virtually unlimited scalability, while the capacity of the on-premises hardware limits DevOps Server. It is important to consider the implications of having your DevOps on premises because of the risks of not getting all the important updates for the tool itself. The cloud-hosted DevOps Services users do not need to update the platform manually because Microsoft handles all the updates and maintenance of it. From the perspective of integration features, DevOps Services provides a more seamless and flexible integration experience due to its cloud-based nature.

Overall, DevOps Services delivers a complete set of tools and services for software development and DevOps. The selection between the two relies on the organization's distinct needs, such as hosting preferences, compliance requirements, and control over the infrastructure.

Azure DevTest Labs for Training, Testing, and Demos

DevTest Labs is a service that allows developers and IT professionals to set up and manage testing and development environments quickly. It provides a fast and easy way to provision and manage various resources, including VMs, containers, etc.

These are the key components and benefits of DevTest Labs:

Rapid provisioning of environments

DevTest Labs allows users to provision environments quickly for testing and development, including preconfigured templates, VMs, and containers. This minimizes the time and effort required to set up testing and development environments and allows teams to get up and running quickly.

Cost savings

Usually, when doing demos and testing, you do not need your virtual machines or servers up and running all the time. Therefore, DevTest Labs provides users with tools and features to manage costs and optimize resource utilization, such as the automatic shutdown of new VMs and cost tracking for each environment.

Security and access control

The Microsoft Entra ID integration, RBAC, and resource locking features help ensure that the testing lab environments you created for your users are secure and accessible only to authorized users.

Integration with other services

DevTest Labs integrates with Azure services, such as Azure VMs, Azure Container Registry, Azure Kubernetes Service, and other tools for integration and automation.

Although the DevTest Labs service has an attractive list of advantages, it also has some limitations worth considering, including the number of VMs you can use, which might not be suitable for production and large-scale testing development environments. Its pricing model depends on the type and amount of resources used within the service.

Across the board, DevTest Labs is a powerful and flexible service that provides users with a quick and explicit way to configure and manage testing and development environments in Azure. This service is useful in training scenarios or if you need a test or demo environment for your small, nonproduction projects.

Overall, DevTest Labs on Azure offers a range of components and benefits, including rapid provisioning, cost savings, access control, integration, and

automation. However, users should know its limitations and potential costs before using the service.

Cloud Development and DevOps with GitHub

Aside from Azure DevOps, Microsoft has an alternative open source tool for DevOps-related tasks for any IT projects for private community, education, and enterprise uses. GitHub is a web-based version control service with a command-line interface (CLI). It is also a DevOps and collaboration platform that enables developers to store, manage, and share their code repositories. GitHub is widely used by many developers, community members, and organizations for open source projects and even for enterprise uses.

Microsoft Azure and GitHub are well integrated and designed to work together in any diverse application development project. They can be used together to provide developers with an end-to-end solution for any application development.

By using GitHub in conjunction with Azure, developers can take advantage of several features and integrations that enable them to streamline their development workflows and deploy their applications more efficiently.

GitHub for Education

In addition to development and DevOps integration with Azure, GitHub can be a convenient tool for teaching development and DevOps practices. It can be used for collaborative coding projects that allow several developers or users to code and collaborate live in the same code workspace. This also enables students to collaborate on coding projects with each other and with their teachers.

GitHub is a good open source tool for teaching version control management, which is an important skill any new developers should know. It also can be used to set up and assess coding assignments. Teachers can

create repositories for each assignment, and students can submit their work by committing their code to their designated repository.

GitHub is known for its open source projects and is a great place to find related open source projects to which students can contribute. By contributing to open source, students can gain experience working with larger and more complex codebases.

Furthermore, an instructor or a trainer can also use GitHub as a platform for developing and sharing curriculum materials such as code examples, assignments, and other learning materials so they are accessible at any time. Generally, GitHub is a potent tool for teaching and learning for everyone, especially for remote and hybrid learning and collaboration.

TIP

Should you choose GitHub or Azure DevOps? Consider GitHub if you are a developer learning Azure or cloud development. Working on open source community projects and collaborating with your small to medium-sized teams are good uses of this service.

On the other hand, you should consider Azure DevOps for enterprise-level projects for large organizations that use Microsoft version controls or if you need to configure and integrate workflows unavailable on GitHub.

Choosing between these two great tools is tough. It is recommended that you keep track of new features and what will be deprecated between them.

Overall, the choice between GitHub and Azure DevOps depends on the size of your team, the complexity of your projects, and your specific needs and requirements. Both platforms have strengths and weaknesses, so it is essential to evaluate your particular use case and determine what is best for your needs.

In **Chapter 14**, development tools such as GitHub CLI, GitHub Codespaces, and other useful tools for cloud development will be discussed.

Cloud Infrastructure Automation and Management

Automating cloud infrastructure involves writing code (e.g., in YAML or Python) to describe the desired state of the infrastructure and using tools to provision and manage infrastructure in a cloud environment automatically.

Infrastructure management refers to overseeing the underlying infrastructure that supports the applications, servers, networking, load balancers, security, monitoring, and other resources that keep systems up and running on the cloud. It typically involves using automation tools and best practices to provision, configure, and manage infrastructure resources, such as VMs or containers. Infrastructure as code (IaC) tools, such as **ARM templates**, **Bicep**, or **Terraform**, can define infrastructure resources and their dependencies as code, making managing and version controlling them easier.

Infrastructure as Code

One of the most common benefits of developing and hosting applications and workloads on Azure is the ability to implement IaC. IaC creates and updates resources such as virtual machines, databases, storage, applications, networking, CDN, and other cloud resources at the infrastructure level. This provides the benefits of repeatable and version-controlled deployment, enabling infrastructure to be treated as a scalable and resilient codebase.

It means you can re-create and redeploy the same resources within that infrastructure in a new instance. The automation tool reads the infrastructure code and uses APIs provided by cloud providers supporting IaC to provision and configure the resources, helping to ensure consistency, reliability, and speed of infrastructure deployment and management.

IaC started as a response to the challenges of managing complex and dynamic infrastructure in the era of cloud computing. As more organizations adopted cloud services, they encountered difficulties manually provisioning and configuring considerable resources and

maintaining consistency and reliability in their infrastructure. IaC emerged as a solution to these challenges by allowing organizations to describe their infrastructure using code and automate the provisioning and management process.

The idea of IaC can be traced back even to the early days of computing, but it gained significant popularity with the rise of cloud computing and DevOps practices where infrastructure can be developed and built as code.

Figure 11-3 illustrates an example workflow of using IaC to build, develop, and manage the infrastructure through design, coding of infrastructure, implementation, and repeating the design process if it does not meet business requirements. Once the design and implementation of the infrastructure are achieved, it can also be updated and improved consistently as new features and requirements arrive.

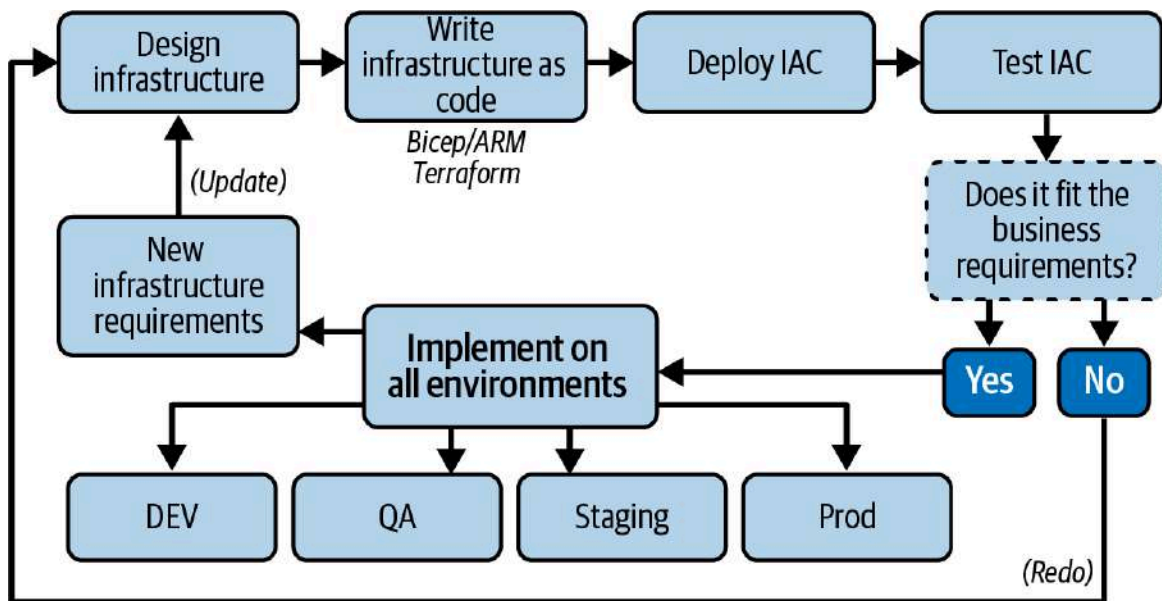


Figure 11-3. Typical infrastructure as code workflow.

By using IaC, organizations can achieve greater agility, scalability, and reliability in their infrastructure and improve collaboration, security, and efficiency. The first IaC tools, such as Puppet and Chef, were released in the late 2000s, and the concept has since become an integral part of modern infrastructure management and cloud deployment.

IaC provides benefits such as improved consistency and reliability on the infrastructure by providing automation and a version-controlled and repeatable deployment process. It also helps diverse teams within an IT project by enabling them to utilize efficient infrastructure provisioning and management. It also motivates collaboration through code quality reviews, version control quality, and many other benefits.

Another benefit that IaC provides is the ability to take control of the security configuration by implementing security policies that help minimize the risks of security threats. Furthermore, adapting IaC in infrastructure automation and management is also beneficial for disaster recovery. This means that as long as the codebase for IaC is saved safely, the infrastructure can be easily re-created in case of failures or unexpected errors in the infrastructure. One of the features of IaC is the ability to quickly and efficiently re-create infrastructure on the cloud, which helps minimize downtime and also assists in ensuring business continuity.

The support for hybrid and multi-cloud deployments is another reason IaC is beneficial to some IT projects and organizations. Using IaC, you can deploy infrastructure across multiple cloud environments, providing greater flexibility and scalability for your organization.

Infrastructure as Code Using Hashicorp Terraform in Azure

Hashicorp Terraform is a widespread IaC tool that can be used to automate the deployment and management of infrastructure resources in Azure. It has gained popularity and many organizations implementing cloud-native solutions on Azure are using it to create, deploy and manage their infrastructure.

Here is how Terraform works with Azure:

Write infrastructure code

Terraform code is written in HashiCorp Configuration Language (HCL), a declarative programming language

typically used to describe the desired state of the infrastructure. For example, when developing IaC code with Terraform on Azure, you can use your desired IDE; however, VSCode is typically used by most developers or infrastructure platform engineers. There are VSCode plugins available that supports IaC languages.

Provision infrastructure

Once the Terraform code is executed using the Terraform CLI, it communicates with Azure APIs to provision and configure resources in Azure. Terraform keeps track of the state of the infrastructure and can make changes as necessary to ensure that the infrastructure is in the desired state. You can store Terraform states on [Azure Storage](#).

Manage infrastructure

When the infrastructure is deployed, changes and updates are expected along the way as the requirements for the systems or applications built in that infrastructure change. Terraform, using the AzureRM provider, can manage the lifecycle of infrastructure resources in Azure, from creation to deletion. Terraform also provides versioning, testing, and collaboration features, enabling teams to collaborate on infrastructure projects.

Connect to Azure

Terraform needs a cloud hosting provider such as Microsoft Azure to provision and manage resources. This can be achieved by authenticating with Azure using a service principal and setting up the required permissions you need to configure and get started with infrastructure development using your preferred choice of IDE locally. You can also use command-line tools that are supported or

integrate your Azure Cloud Shell with the Terraform Azure provider.

By utilizing Terraform and IaC with Azure, organizations can achieve greater efficiency, consistency, and reliability in their infrastructure while reducing manual effort and improving collaboration.

Infrastructure as Code Using Azure Resource Manager and Bicep

Azure Resource Manager (ARM) is a deployment and management service in Microsoft Azure that enables organizations to provision and manage resources in Azure. ARM uses JSON templates to define the desired state of the infrastructure and provides a set of REST APIs for programmatic management of the resources.

These JSON files are also known as ARM templates. Like Terraform, an ARM template is based on a declarative syntax. It is a process of specifying the end configuration without focusing on defining how the task should be completed.

An ARM template contains five components:

Parameters

The template allows you to provide values at deployment time so you can reuse your templates for different Azure subscriptions or tenants easily. As a result, parameters improve the reusability of your templates.

Variables

In variables, you can specify values you can reuse throughout your template. The difference between variables and parameters is that the values of variables are typically

defined by the template author and not changed, and not provided during deployment.

Functions

You can define custom functions. They are expressions where you can define the logic of your template.

Resources

This is where you specify the declarations of all the resources you want to deploy through your template.

Outputs

This section allows you to return values upon completion of the template deployment process. It is useful for retrieving information about the resources deployed.

ARM templates are a compelling asset in your toolkit for Azure. The downside is not the concept of ARM templates itself but the JSON language you need to master to create them. The JSON language is great for the ARM engine to process IaC. It is also a good machine language for exchanging data, but authoring and reading JSON code is difficult. The JSON language has a lot of syntax overhead and is hard to read, author, and debug. Microsoft realized this as well, and as a result, the Bicep language was born. Bicep is a domain-specific language (DSL) and similar to ARM templates specifically designed for the Azure Cloud.

Azure Bicep is a new, simpler IaC language for Azure resource deployment. Bicep typically supports a simpler and more readable syntax than JSON, making it easier to use and learn for organizations starting with IaC. It acts as a transparent abstraction layer on top of ARM templates, improving the authoring experience of IaC. It has a much cleaner syntax. Bicep also improves type safety and has much better support for modularity to reuse your code.

Because it is a transparent abstraction layer on top of ARM and ARM templates, there is full feature parity. Anything you can do with ARM templates can also be done using API versions, and properties valid inside ARM templates are also valid in Bicep.

Upon deploying a Bicep template to Azure, it is transpiled (taking source code written in one language and transforming it easily into another) into an ARM template. The ARM template becomes an intermediate language (IL) before it is handled by the ARM engine in Azure as part of your deployment.

So what are the differences between the ARM and Bicep languages? They include:

Syntax

ARM uses JSON templates to define the infrastructure. In contrast, Bicep uses a simplified syntax designed to be more readable and easier to use.

Integration

ARM is deeply integrated into Azure and provides a comprehensive resource deployment and management solution. On the other hand, Bicep is designed to be a more straightforward IaC language specifically for Azure resource deployment.

Features

ARM provides rich features and capabilities for resource deployment and management. At the same time, Bicep offers a more straightforward set of features focused on making it easier to use and learn.

Compatibility

ARM templates are widely used and supported in Azure and can deploy many resources. Bicep is compatible and can be

extended with the Kubernetes provider. By doing this, you will be able to create resources that allow Kubernetes to interact with Bicep directly using a Kubernetes manifest file and the Kubernetes CLI (kubectl).

Migration from ARM templates in JSON format to the Bicep language is also supported. See Microsoft's [five-phase migration flow](#) on how to do this.

Organizations can choose between ARM and Bicep based on their needs and preferences. Bicep provides a comprehensive and feature-rich resource deployment and management solution with full feature parity for Azure resource deployment.

When to Consider Azure Terraform over Azure Bicep or ARM

Terraform and Azure's Bicep and ARM are all IaC tools that can automate the deployment and management of infrastructure in Azure. However, each service has strengths and is better suited for different use cases.

Here are some common use cases for using Terraform instead of Bicep:

Multi-cloud infrastructure deployments

Terraform is a multi-cloud tool that can manage infrastructure in multiple cloud providers, including Azure. This makes it a good choice for organizations that need to deploy infrastructure in numerous cloud environments.

Large-scale deployments

Terraform provides a scalable solution for managing large-scale infrastructure deployments that support different types of cloud providers, which is ideal if your project is aiming for multi-cloud use cases. It allows users to utilize various resource types and offers Azure providers

(AzureRM), which makes it suited for complex deployment scenarios. For example, Terraform needs a state or state file to operate, a “database” that maps Terraform configuration files to the cloud on Microsoft Azure. Upon deploying, Terraform reads the current state of any existing object or Azure resources. Then it compares the current configuration to the initial state and proposes the change actions that match the sources to the configuration file. In the Terraform plan stage, you can see what resources or objects will be created, updated, or destroyed beforehand. On the other hand, Azure Bicep is a newer, simpler IaC tool designed explicitly for Azure resource deployment. It provides a simplified syntax and is designed to be easier to use and learn, making it a good choice for organizations starting with IaC or preferring a more straightforward solution.

Although the state file Terraform uses has advantages, you have an additional file to manage and maintain that becomes critical to your deployments. Bicep does not use a state file because, to Bicep, Azure is the state. Moreover, Bicep is also better integrated with Azure, providing a seamless experience for deploying and managing Azure resources.

TIP

Azure Export for Terraform is a good tool to consider if you are trying to migrate and translate your infrastructure resources to Terraform.

And lastly, there can be a delay in getting new resources or resource APIs available in Terraform. Someone from the community must first update resources to the Azure Terraform provider. The Terraform community is, however, quite active, so it usually does not take that long. On the other hand, Bicep always has day-0 support for new resource types or new API versions.

It is not Microsoft's intention to make Bicep the language that rules them all. Microsoft does not want to compete with Terraform or any other language. Microsoft also has a dedicated team to improve Terraform for Azure continuously.

All languages have pros and cons, and having a choice is great. There is no reason to change if you have already invested in learning Terraform. Overall, you need to choose the IaC language that suits your requirements and that you feel most comfortable with.

Configuration as Code

Configuration as code (CaC) refers to managing infrastructure and application configurations through source code rather than manual configuration through a web interface or administrative tools. This allows for version control, review, and collaboration when making changes to infrastructure and applications, as well as improved automation and consistency.

There are several benefits to using CaC, including:

Version control management

The changes implemented by infrastructure engineers or developers can be monitored, managed, and tracked. Version control management also enables easy rollback of mistakes or errors when necessary.

Reusability of configuration and settings

Configurations can be easily re-created and reused, reducing errors and increasing reliability.

Automation to multiple environments for consistency

Configurations can be automated and managed consistently across multiple environments.

Enable collaboration

Teams in different roles and responsibilities can quickly review and modify configurations, enabling them to collaborate and communicate more often.

Auditing and monitoring

All configuration changes are recorded and can be audited if necessary.

Support for scalability

Configuration management can be easily scaled, especially for those systems with infrastructure that continue to grow.

Overall, CaC improves efficiency by helping your teams collaborate using automation and consistency in application configurations and infrastructure management. It also helps in reducing the risks of human errors and mistakes in the configuration process.

Policy as Code

In addition to IaC and CaC, there is policy as code (PaC), which can be used with cloud solutions and applications in Azure. PaC refers to defining policies for resources in Azure using code. This allows administrators to enforce rules and best practices for resource management and security in a more automated and scalable way.

With PaC, policies are defined using declarative language in a format like JSON or YAML and stored in a version-controlled repository. These policies can then be applied to Azure resources to ensure they adhere to specific standards, such as requiring certain tags or disallowing insecure protocols.

In Azure, PaC is implemented using Azure Policy, a service that provides users or administrators a way to create, assign, and manage policies across a range of Azure resources. Administrators can use built-in or custom policies to meet specific business requirements.

A typical policy requirement is enforcing tagging of Azure resources. It ensures all Azure resources are tagged with specific metadata, such as environment, owner, and cost center. You can use PaC to define a policy that enforces this tagging requirement on your Azure subscription's new and existing resources. Azure Policy is a good tool for implementing these within your resources in Azure.

Another everyday use case for PaC is to enforce security controls, such as requiring encrypted storage or prohibiting unsecured protocols. To implement this policy, you would define a JSON or YAML policy file that specifies the required security controls and then create an assignment to apply the policy to your Azure resources.

One of the reasons enforcing policy is important is because of compliance. Organizations in regulated industries may be required to comply with specific regulations, such as HIPAA in the medical field or PCI DSS in industries where credit card data is handled. PaC can enforce compliance with these regulations by defining policies ensuring that all Azure environment resources meet the required standards.

For example, you could create a policy prohibiting sensitive data storage in unencrypted storage accounts. By implementing policies within your infrastructure and its resources on Azure, you and your organization create rules and configurations that fit the organization's compliance, data privacy, security, and identity protection and manage the resources being used.

By using PaC in Azure, organizations can enhance their governance and compliance measures, lower the risk of security violations, and gain more important consistency and standardization across their cloud environment.

Azure provides several built-in policies through Azure Policy that can be used as templates or starting points for creating custom policies. You can also use third-party tools and frameworks like Terraform to manage **PaC in Azure**.

NOTE

You may have heard the term policy as code (PaC) used similarly to configuration as code (CaC). While PaC and CaC are both practices for automating and managing infrastructure in modern software development, they have different goals and purposes.

PaC codifies organizational policies, infrastructure, security, and compliance guidelines into machine-readable and executable code. It ensures that infrastructure complies with regulatory requirements, security standards, and internal policies. With PaC, developers can write code to test and enforce policies, which helps ensure that infrastructure configurations meet security and compliance requirements.

In contrast, CaC refers to writing code to define and manage infrastructure configurations. CaC is used to automate the provisioning and management of infrastructure resources such as VMs, networks, storage, and other services. CaC helps ensure that infrastructure is consistent and reproducible, reducing the risk of misconfigurations and improving the reliability and scalability of systems.

In summary, PaC focuses on enforcing policies and guidelines, while CaC focuses on automating the configuration and management of infrastructure resources. Both practices are essential for software development and are often used to ensure that infrastructure is secure, compliant, and reliable.

Monitoring and Infrastructure Management in Azure

Now that you know the different tools for infrastructure management, next you'll learn about the different Azure monitoring tools used to monitor the infrastructure. Monitoring tools provide and help organizations facilitate their IT operations, reduce costs, and improve their cloud infrastructure's performance, reliability, and security.

Azure Monitor

Intelligent monitoring across Azure can monitor how your applications and Azure resources are doing. Azure Monitor is a centralized tool that helps you effectively monitor the availability of your applications and services. Using Azure Monitor will give you metrics and tracking information that will provide real-time visibility into the health and performance of Azure. It can be configured to send alerts in different methods and even integrate with other event-driven services within Azure or external ones for proactive resolution and actions when issues impact your application's performance or even your users in the production environment.

Application Insights

As a subcomponent of Azure Monitor, Application Insights is ideal for application metrics and telemetry monitoring. It can observe the infrastructure and identify what issues are impacting your applications, including your users.

Azure Log Analytics

If you and your team need to log performance data from your applications and infrastructure, then you can do that using Azure Log Analytics. This tool can gather all the data you need to identify and troubleshoot issues impacting your cloud resources within your infrastructure.

Azure Network Watcher

Networking is essential when your workloads are on Azure. Therefore, a suitable tool such as Azure Network Watcher can help you monitor how networking infrastructures are doing. The Network Watcher service will help you monitor, diagnose, and visualize the networking architecture that supports and connects your applications running on the

cloud. It also helps you identify any possible issues and security risks within your networking configuration, allowing you to improve them.

Azure Advisor

This Azure service is intelligent and AI powered to help provide customized recommendations based on your current cloud resources. Azure Advisor is like an assistant that provides the best possible recommendations, remediations, and actions you need to help you improve your infrastructure or services on Azure. It can guide you in optimizing your Azure resources, such as possible recommendations on improving performance, reducing costs, and improving security.

Figure 11-4 shows an example of how Azure Monitor is implemented when observing Azure Kubernetes Service containers. Metrics and logs needed for monitoring, performance, and diagnostics are being collected for reporting, insights, or alerts based on events to be monitored.

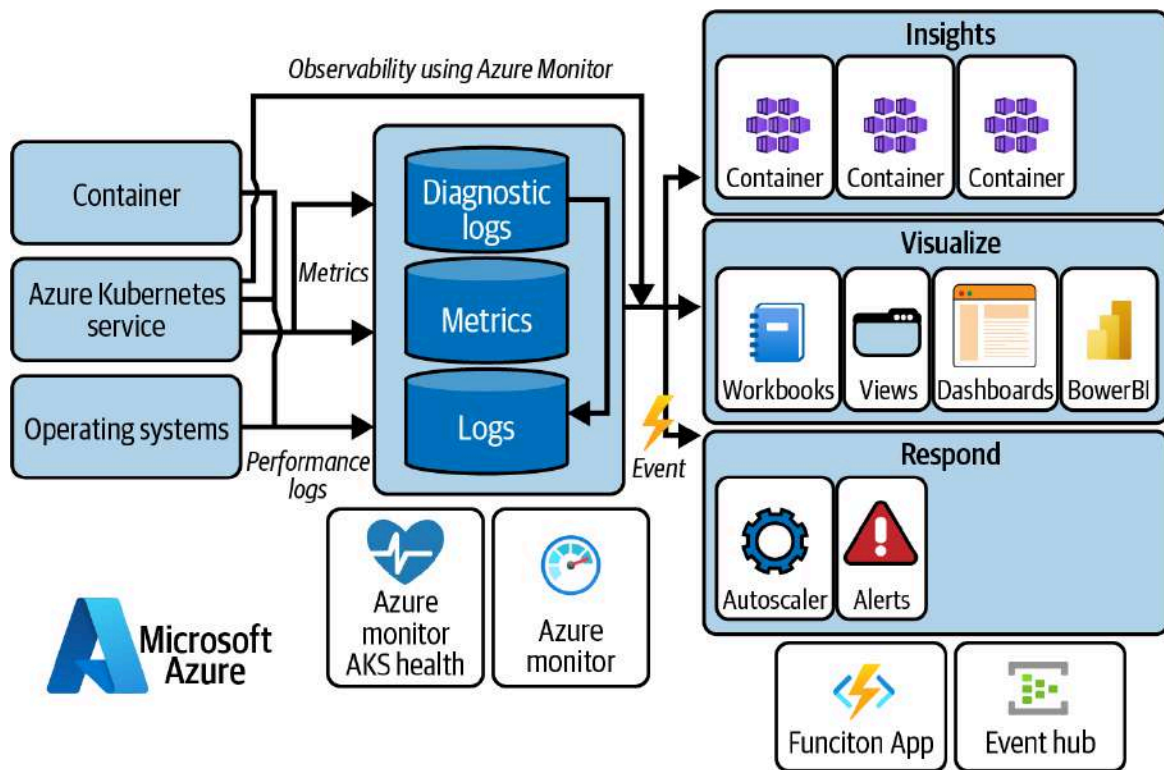


Figure 11-4. Monitoring Azure Kubernetes Service with Azure Monitor

Overall, monitoring tools are helpful in observability, which gives you and your team a comprehensive picture of your cloud infrastructure and the resources running on it. Monitoring infrastructure helps ensure that your applications are performing well and identifies potential opportunities to improve your applications for the benefit of the users.

Learn By Doing (Try It!)

The following are recommended tutorials from Microsoft's official documentation.

- [Create a CI/CD pipeline for GitHub repo using Azure DevOps](#)
- [Deploy ASP.NET Core App to Azure Kubernetes using Azure DevOps Starter](#)
- [Learning series of exercises using Azure DevOps Labs](#)

- [Quickstart: Create a policy assignment to identify non-compliant resources using Terraform](#)
- [Learning Path: Deploy Azure resources by using Bicep and Azure Pipelines](#)
- [Microsoft Learn: Build your first Bicep template](#)

Summary

In this chapter, you learned more about cloud infrastructure services in Azure that enable you to build, deploy, and manage the application lifecycle using DevOps tools, practices, and more.

Azure DevOps provides continuous integration, delivery, and application lifecycle management tools. It allows teams to collaborate and automate the entire software delivery process from code development to deployment.

DevTestLabs is a service in Azure that enables users to provision development and test environments in the cloud quickly. It helps users reduce the time and effort required to manage their infrastructure for testing and development purposes.

Infrastructure as code (IaC) operates and provides infrastructure automation through code instead of manual processes. This approach allows users to automate their infrastructure and reduce the risk of configuration errors and inconsistencies. It also provides benefits such as flexibility and minimizes the risks of cloud vendor lock-in.

Policy as code (PaC) is a service that allows users to define policies in code that govern the configuration and behavior of their cloud resources. This approach helps users enforce compliance, security, and governance across their infrastructure.

We also learned that Configuration as Code (CaC) is about managing infrastructure and application configurations by code instead of configuring settings manually. Combining PaC, CaC, and IaC is commonly called GitOps, which is often associated with DevOps.³

Azure services for monitoring and observability are available, such as the Azure Monitor, Application Insights, Log Analytics, Azure Network Watcher, and other tools that monitor and protect your Azure resources.

In **Chapter 9**, you learned about the different services you can implement in your infrastructure to protect your workloads, resources, data, and users on Azure using Microsoft Defender for Cloud and other defenders for specific environments and categories.

Overall, these monitoring tools help users efficiently provision and manage their infrastructure in Azure, deploy processes, and enforce compliance and governance policies.

Check Your Knowledge

1. How does Azure DevOps streamline the software development lifecycle in the cloud?
2. How can you use Azure Application Insights to monitor the performance and availability of your applications?
3. What are the benefits of using Azure DevTest Labs for testing and deploying applications in the cloud?
4. How can you use Azure Policy to enforce governance and compliance standards across your cloud infrastructure?
5. What are the benefits of using Azure Network Watcher to monitor and diagnose your cloud network infrastructure issues?

For the answers to these questions, see the **Appendix**.

Recommended Learning Resources

“Azure DevOps.” Microsoft Azure products, <https://oreil.ly/kcVz->.

“Azure Policy Documentation.” Microsoft Learn, <https://oreil.ly/EV8bf>.

Berson, Freek. *Getting Started with Bicep: Infrastructure as Code on Azure*. Self-published, 2021.

Brikman, Yevgeniy. *Terraform: Up and Running (Third Edition)*. Sebastopol, CA: O'Reilly Media, 2022.

Davis, Jennifer, and Ryn Daniels. *Effective DevOps*. Sebastopol, CA: O'Reilly Media, 2016.

“Design Azure Policy as Code Workflows.” Microsoft Learn, April 4, 2023, <https://oreil.ly/I3h5->.

“Infrastructure as Code.” Microsoft Learn, May 22, 2023, <https://oreil.ly/UfE9R>.

Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook*. Portland, OR: IT Revolution Press, 2016.

Morris, Kief. *Infrastructure as Code (Second Edition)*. Sebastopol, CA: O'Reilly Media, 2020.

Sharma, Sanjeev. *The DevOps Adoption Playbook*. Indianapolis, IN: Wiley, 2017.

“Terraform on Azure Documentation.” Microsoft Learn, <https://oreil.ly/kCX4J>.

¹ Harrison Clarke, “How *DevOps Engineer* became the most in-demand job title,” blog post, September 28, 2021, <https://www.harrisonclarke.com/blog-2023/how-devops-engineer-became-the-most-in-demand-job-title>

² Lori Perri, “What is Platform Engineering?” Gartner, October 5, 2022, <https://www.gartner.com/en/articles/what-is-platform-engineering>

³ “What is GitOps?” GitLab, <https://about.gitlab.com/topics/gitops>

Part V. Governance, Migration, Architecture, and Development Tools

The final part contains chapters that covers topics such as cloud infrastructure management, compliance, and governance on Microsoft Azure. This part also tackles topics about cloud migration scenarios and different tools you can use to be successful in your organization's cloud adoption, innovation, and migration journey, regardless if you are hosting fully on cloud, hybrid, or multi-cloud. It helps broaden your knowledge on the Cloud Adoption Framework and Well-Architected Framework for Microsoft Azure.

This part ends with useful developer tools, technologies, and advice that will help developers who are working with and building solutions with Azure as a cloud platform.

Chapter 12. Cloud Management and Governance in Azure

In a Cloud Journey, before building any migration strategy and talking about cloud migration, it is important to govern the environment and workloads. In an on-premises environment, organizations already have their policies and rules that can be optimized and completed by Cloud governance to pursue guiding it in the cloud operations. We cannot guarantee the success of adoption and ensure that data security, system integration, asset deployment, and cloud computing are properly managed if it is not governed because we may encounter unexpected problems. We must align cloud adoption with cloud governance functions or the creation of a new team that deals with governance. Commitment to cloud governance ensures that a complex hybrid environment meets organizational policies, compliance obligations, and security best practices.

— Hamida Rebai, Cloud Solutions Architect, Microsoft
MVP and MCT, Docker Captain

This chapter dives into how Azure delivers a range of services ideal for cloud management and governance. You will learn some of the services in Azure that can help with cloud management, administration, compliance, and learning FinOps in the cloud for cost optimization.

Cloud Infrastructure Management and Governance

Cloud management and governance are essential for managing cloud resources and ensuring compliance with regulatory standards, organizational policies, and best practices. There are many different use cases for cloud computing technologies in Azure in various industries.

For example, an aerospace company may use Azure to power its airplane design, development, and analytical tools. The scalable and flexible platform enables the aerospace company to process massive amounts of data, which helps the company make better decisions and reduce costs. If you'd like to learn more about how Boeing is leveraging the Microsoft Cloud and its AI capabilities to update its technology infrastructure, see [this article](#).

Healthcare companies can use Azure to build and deploy cloud-based solutions for healthcare providers. These solutions help healthcare providers manage patient data, optimize workflows, and improve patient outcomes. Azure is also used for technical solutions, such as advanced medical imaging. See this [case study](#) on GE Healthcare using Azure to improve medical diagnostics.

Automobile manufacturers also use Azure, which has capabilities that can help develop and deploy cloud-based connected car platforms. Using cloud computing, a company can collect data from its cars and use it to improve car performance, develop new features, and provide better customer service. One example of the implementation of these modern features is Mercedes-Benz using the [Azure OpenAI](#) service to enhance the driving experience of its customers.

These examples show how Azure is helping businesses and organizations innovate, improve efficiency, and reduce costs by leveraging cloud computing capabilities. However, these improvements are only possible with proper and strategic cloud infrastructure management.

Let's look at why management and governance in the cloud are essential:

Effective management of cloud resources

Cloud infrastructure management enables organizations to manage their cloud resources more effectively, ensuring that they're deployed and configured correctly and being used efficiently. This includes monitoring performance, troubleshooting issues, and scaling resources up or down as needed.

Complying with regulatory standards

Cloud compliance and governance ensures that an organization's cloud resources are compliant with regulatory standards such as GDPR, HIPAA, and PCI DSS. Organizations can avoid costly penalties and protect sensitive enterprise and user data by implementing compliance policies and procedures.

Implementing consistent organizational policies

Cloud infrastructure management means managing its governance. Doing this makes an organization's cloud resources consistent and aligned with administrative guidelines. Standard corporate policies include security controls, data retention, and cost management practices. Organizations can reduce risk, maintain consistency, and optimize costs.

Optimized cloud cost management

This helps organizations optimize their cloud costs by monitoring usage, identifying underutilized resources, and implementing cost-saving measures such as reserved and spot instances.

Enhanced security and risk management

When organizations use security controls and risk management practices, they can identify and mitigate security risks. This includes monitoring for vulnerabilities and threats and enforcing access controls, encryption, and other security benchmarks.

Disaster recovery management

cloud management and governance and governance practices are critical for disaster recovery and business continuity planning. By implementing backup strategies and disaster recovery procedures, organizations ensure that their cloud resources can be quickly restored during a disaster and that critical applications and data are always available to users.

As organizations adopt multiple cloud platforms to meet their business needs, cloud infrastructure management and governance practices become increasingly important. By establishing standardized policies and procedures, they can ensure that their cloud resources are managed consistently across all platforms and that they're being used effectively and efficiently.

Furthermore, implementing exemplary practices can automate routine tasks, such as deployment, configuration, and maintenance of cloud resources. This can help organizations improve operational efficiency, lower the chance of human error, and ensure consistent application of policies and practices.

Overall, handling governance in the cloud is critical for organizations leveraging cloud resources. By implementing best practices, policies, and tools for cloud management and governance, organizations can ensure that their cloud resources are used effectively and efficiently while maintaining compliance with regulatory standards and organizational policies. The following sections explore how to do this.

Azure Resource Manager

Azure Resource Manager (ARM) is a powerful tool that enables organizations to manage and deploy their cloud resources efficiently. Using ARM, an organization can create, update, and delete resources in a single, consistent way, using declarative templates that define the desired state of the resources. ARM also makes managing complex cloud environments

easier while improving the security and compliance of the resources. They can do these through security, identity, and compliance management tools available from administrator to administrator using Microsoft Entra ID.

ARM provides a unified and consistent approach to managing resources across different Azure services and allows organizations to automate many everyday tasks, such as deployment, configuration, and scaling of resources.

Additionally, with the implementation of ARM, organizations can more easily manage their cloud resources while improving their security, compliance, and operational efficiency.

This is possible because ARM provides a unified view of all resources and enables organizations to handle them with templates and automation. This helps organizations to manage and structure resources, apply policies in bulk, and ensure compliance with organizational standards.

Figure 12-1 illustrates how ARM helps manage Azure resources in the platform. Azure users can create resources differently through Azure SDKs using command-line tools such as Azure PowerShell and Azure CLI. Those that prefer to do it manually with a user interface can also manage the resources using the Azure Portal.

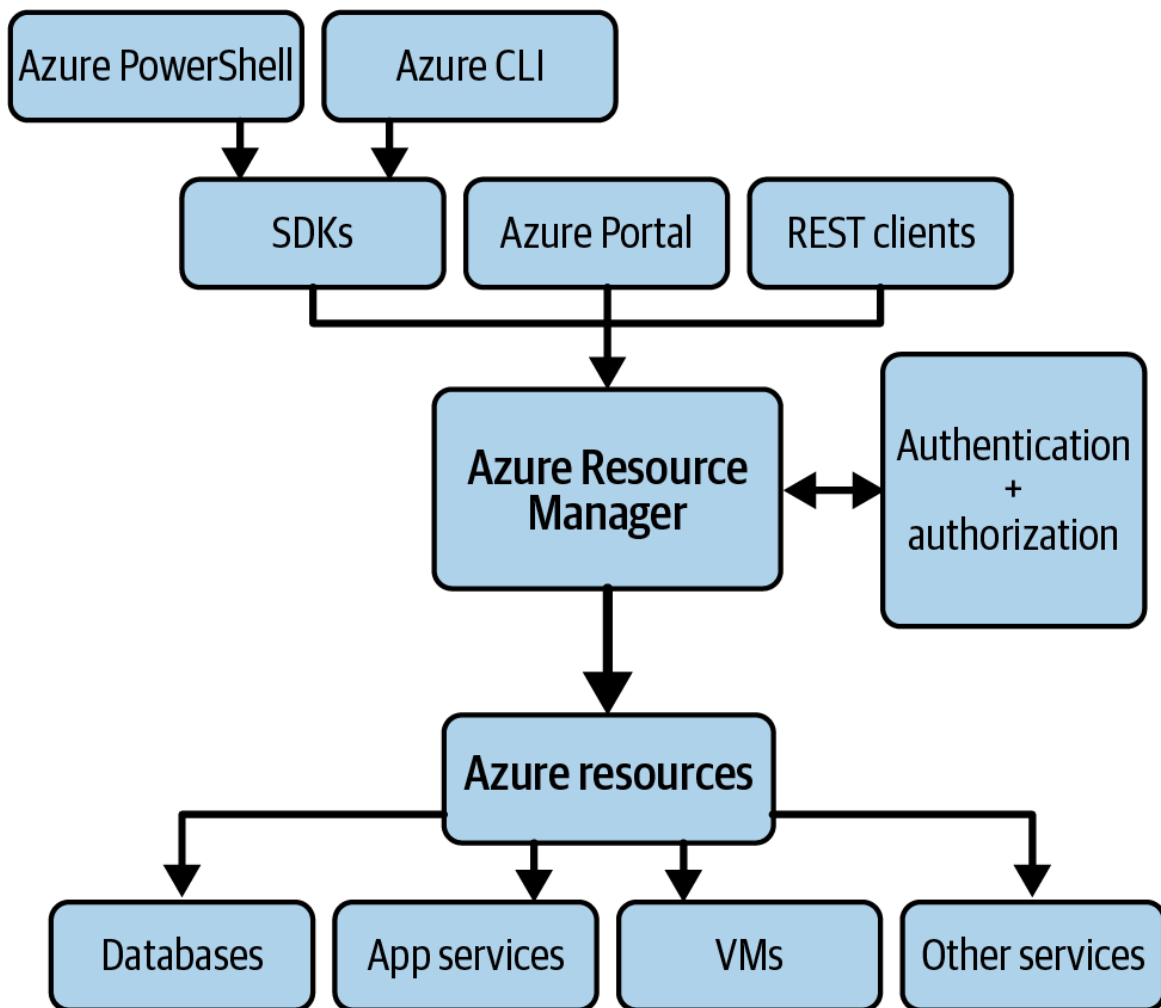


Figure 12-1. ARM and its associated Azure resources, Azure SDKs, and platforms

REST API clients can also use ARM to create, modify, and manage Azure resources. These processes have an added layer of authentication and authorization for security.

To start using ARM for managing resources in Azure, you should be familiar with how deployment and management services for Azure resources work in the cloud. ARM provides a way to deploy, manage, and organize resources using templates, role-based access control (RBAC), and resource groups.

Since you already have an Azure account and have created a resource group for your cloud resources in Azure, you can export the ARM template of that creation and reuse it to deploy it again.

You can deploy resources to Azure using an ARM template, a JSON file that defines the infrastructure and configuration of the resources you want to deploy. You can create your template or use one of the many templates available in the [Azure Marketplace](#) and [Microsoft Learn](#).

Once your resources are deployed, you can manage them using the Azure Portal, Azure CLI, Azure PowerShell, or REST APIs. You can view resource properties, scale resources up or down, or delete resources when you no longer need them.

Aside from deploying resources using an ARM template, it also helps you manage and automate infrastructure deployments using infrastructure as code (IaC) through Azure Bicep or Terraform, as discussed in previous chapters.

You can also authenticate resource management using Azure. Using Azure RBAC to control access and permissions on Azure allows you to control who can access resources in your subscription and what they can do with those resources. You can designate roles to groups, users, or applications and define permissions for those roles.

Managing and Organizing Resources Using Azure Resource Groups

One of the components of ARM is the use and implementation of the logical container called a resource group. Think of this as a virtual container on Azure where you can organize and manage resources. In other words, a resource group is a logical container for resources deployed, managed, and shared across different IT teams.

Let's look at why using resource groups in Azure helps deploy, group, and manage resources as a single unit:

Organize Azure resources

You can group resources by their function, application, environment, or any other logical grouping, making it easier

to find and manage resources and understand their dependencies.

Access control management

Your organization and Azure administrators can control resource access by defining roles and assigning permissions to users or groups. You can also set up access policies to enforce governance and compliance requirements.

Monitor resources for improvement

Monitoring for performance and reliability is essential to keep workloads robust and secure. On Azure, you can monitor the health and performance of Azure resources within a resource group and set up alerts and notifications for specific conditions. This allows you to identify and resolve issues before they impact your business.

Automate resource management tasks

Automating repetitive and manual tasks is a strategy for efficiency and time management in many IT organizations. You can use automation tools such as Azure Automation, Azure Functions, and Azure Logic Apps to automate management tasks for your resources within a resource group. Automation tools can be used to deploy resources to your resource groups. This can reduce manual effort and increase the efficiency of your operations.

Cost management by resource group

As your organization expands, so do resources and workloads you host on Azure. Keeping track of monthly costs for different categories and workloads can be difficult without the help of Azure resource groups. They can help you track the prices of resources and you can use cost

management tools to optimize your spending. This can help reduce waste and improve your return on investment.

As you can see, using resource groups to manage resources in Azure provides several benefits, including improved organization, security, monitoring, automation, and cost management. By grouping resources and ordering them as a single unit into a resource group, you can simplify your management processes, improve efficiency, and enhance your overall governance and compliance posture.

Azure Resource Locks for Cloud Assets Protection

Azure resource locks are IT infrastructure cloud assets because they allow organizations to protect their infrastructure resources from being deleted or changed accidentally. DevOps engineers and Azure administrators can use Azure locks to protect app services, storage, databases, etc., from unintended events, regardless of the permissions the user of these resources has.

Users can enable Azure locks manually on the Azure Portal or automate them using IaC ARM, Bicep, Terraform, and other tools. Following are the Azure lock settings you can use to protect your resources.

ReadOnly

Only users with subscriptions or authorized tenants can read resources with this type of lock. This configuration will block users from deleting and modifying your resource. ReadOnly lock will prevent all users from accessing the permissions that the RBAC role reader provides.

CannotDelete

This option allows authorized users to modify or read Azure resources but they cannot delete them. It is important to note that when you apply this lock, it actually prevents the deletion of Azure RBAC assignments on a resource or

resource group. Applying this lock at the resource group level will also cause your backups on the Azure Backup Service to fail.

Locks follow the parent scope, meaning that if you add a lock on a resource group, the resources under that group will inherit the same lock. If both locks are enabled, the most restrictive lock precedes resource inheritance. You must also consider security interactions, for example, a lock that cannot be applied in a network security group because it will affect and block traffic flows. Read about these and other important **considerations** before applying locks to your Azure resources.

Azure Blueprints (Preview)

Blueprints are predefined templates and configurations that allow developers and organizations to quickly and efficiently deploy cloud environments that adhere to organizational standards, policies, and compliance requirements. They serve as instructions that automate the deployment of standardized cloud resources and services. You can also manage blueprints as code if you want to integrate with CI/CD release pipelines on Azure DevOps or if you want to store them in a source control repository.

Azure Blueprints predefines the infrastructure of an organization's environments with management groups specific to different departments' IT, marketing, development, and infrastructure teams. Each management group has other Azure subscriptions associated with them. Each blueprint has a set of related artifacts, such as Azure RBAC, policies, resource groups, etc.

Cloud infrastructure or DevOps engineers can use Azure Blueprints to create and deploy cloud infrastructure that follows best practices, industry standards, and regulatory compliance requirements.

AZURE BLUEPRINTS (PREVIEW) DEPRECATION IN 2026 AND MIGRATION RECOMMENDATION

As per Microsoft updates, Azure Blueprints is due for deprecation. It is recommended and advised to migrate existing definitions and assignments of Azure Blueprints to **deployment stacks** and **ARM template specs in Bicep**.

Aside from providing a standardized approach to cloud deployment, Azure Blueprints also offers the following benefits for businesses:

Faster time-to-market

Azure Blueprints allow users to quickly deploy standardized cloud environments, reducing the time it takes to get applications and services to market.

Improved security

By enforcing specific policies and configurations, blueprints help ensure that cloud environments are secure and comply with regulatory requirements.

Simplified management

With blueprints, businesses and organizations can manage their cloud environments more efficiently by automating the deployment of resources and services.

Blueprints help businesses and organizations ensure consistency and compliance across cloud environments. They define and enforce policies and configurations across cloud resources, ensuring that deployments meet organizational standards and regulatory compliance requirements.

Creation and Deployment of Azure Blueprints

Azure Blueprints provides a structured approach to managing cloud infrastructure by automating the deployment of resources and policies. The

implementation and lifecycle of cloud governance and infrastructure blueprints have steps and processes that can iterate. Creation and management of blueprints involves:

Defining blueprint scope

This can be done by selecting the Azure subscription, resource group, or management group where it will apply.

Creating a blueprint definition

After defining the scope, create a definition by specifying the name, description, and version. The definition also includes the artifacts that define the blueprint, such as policies, resource groups, and templates.

Adding artifacts

Artifacts are added to the blueprint definition and used to define the blueprint's infrastructure and policies. These can include ARM templates, policy definitions, and role assignments.

Publishing the blueprint

When the blueprint is published, it becomes available in infrastructure deployments.

Assigning the blueprint

Set the desired scope of the blueprint, e.g., subscription, resource group, or management group.

Tracking blueprint assignments

The next critical step in the blueprint lifecycle is tracking the status of blueprint assignments to ensure that the configuration is correctly applied to the specified resources.

Updating the blueprint

When needed, update the blueprint definition to modify the configuration of already defined resources and policies.

By following these steps in its lifecycle, organizations can ensure that their cloud infrastructure is configured consistently and compliantly across all deployments.

Azure Blueprints for Zero Trust Security and Cloud Migration

Cloud adopters use the Zero Trust methodology for cloud security management. When it comes to Zero Trust security, Azure Blueprints provides several benefits:

Standardization

Blueprints help organizations enforce standard security policies and configurations, and as a result, they help reduce the risk of security breaches.

Ensure consistency

By defining a set of preconfigured resources and configurations, Azure Blueprints ensures that all resources deployed within an Azure environment follow the same security policies and configurations. If used for the cloud migration process, it will verify that Azure resources being created and deployed during migration follow the same structure and policies, which can reduce the risk of errors and misconfigurations.

Infrastructure automation

Blueprints enable organizations to automate the deployment of security controls, making it easier to manage and enforce security policies at scale. Automation capabilities using blueprints also help organizations with cloud adoption by reducing the time and effort required for migration.

Security risk auditing

Using Azure Blueprints allows organizations to audit their Azure environment to ensure all resources are deployed and configured correctly, identifying potential security gaps.

Simplify the cloud migration process

Using Azure Blueprints can help define a set of preconfigured resources and configurations required for a specific migration scenario, making migrating applications and workloads to Azure easier.

Blueprints can help organizations achieve a more secure and streamlined cloud migration process while ensuring their Azure environment adheres to organizational standards and best practices.

Azure Monitor for Monitoring and Reliability

As the name implies, Azure Monitor provides tools for monitoring the performance and health of Azure resources. It enables organizations to gain insights into the health of the applications and infrastructure and proactively identify and solve issues before they become a concern.

Azure Monitor is a cloud-based monitoring service provided by Microsoft Azure that helps developers and IT professionals gain insights into the performance and health of their applications and infrastructure deployed on the Azure platform. It offers a comprehensive set of tools and features for monitoring, analyzing, and visualizing telemetry data from various sources, including applications, services, and VMs.

Benefits of Azure Monitor include:

Centralized monitoring

Azure Monitor provides a centralized platform for monitoring multiple Azure resources and services. This helps to streamline monitoring operations and reduce the time and effort required to monitor multiple resources.

Real-time insights

Monitor offers real-time monitoring capabilities that enable developers and IT teams to quickly identify and diagnose issues before they become critical. This helps ensure that applications and services are always up and running, providing users with high availability.

Customizable dashboards

Azure Monitor allows users to create custom dashboards that display essential metrics and KPIs related to their applications and services. Users can visualize their resources' health and performance in a single view.

Monitoring alerts

Azure Monitor provides alerting capabilities that enable users to set up alerts based on specific metrics and thresholds. Users are notified immediately when an issue occurs, allowing them to take corrective action quickly.

Figure 12-2 describes how Azure Monitor collects valuable data such as metrics, logs, changes, and traces from its source applications, infrastructure, and resources. This collected monitoring telemetry can then be analyzed using Azure Log Analytics. You can also visualize the collected data using Power BI, Grafana, Workbooks, etc.

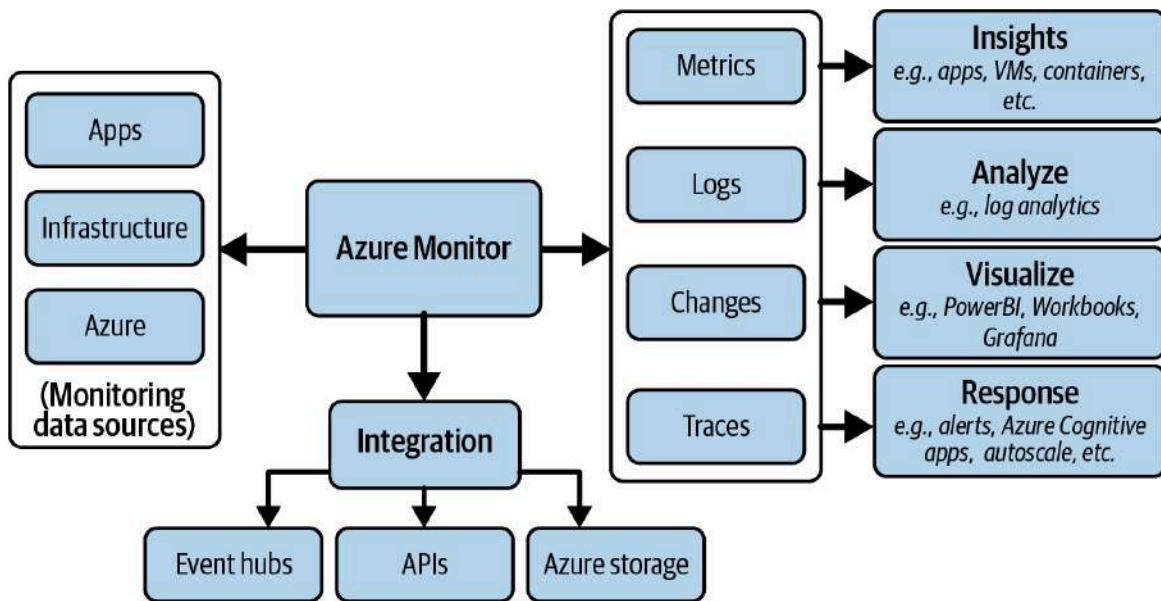


Figure 12-2. Overview of Azure Monitor

If you want to monitor and get alerts about specific monitoring events in your applications or infrastructure, you can use alerts, logic apps, serverless computing services, and other monitoring tools for your Azure workloads.

Overall, the monitoring and insights provided by Azure Monitor help identify and diagnose issues, optimize performance, and improve the overall health and availability of your applications and services.

NOTE

The scope of this book does not cover all of the concepts of Azure Monitor and its components; however, if you want to learn more about the structure of this widely used monitoring service on Azure, check out the [documentation](#).

Azure Automation

Azure Automation is a cloud-based service used for automation management for repetitive and time-consuming tasks. It is designed to help developers and IT professionals streamline workflows and reduce manual efforts.

This service provides tools for automating the management of Azure resources, which removes time-consuming and tedious manual processes. Using automation in Azure, organizations can automate tasks such as deployment, configuration, and maintenance of Azure resources.

Figure 12-3 illustrates that Azure Automation is a heterogeneous, hybrid, cross-platform solution enabling features such as configuration management and orchestration of processes using Python runbooks or scripting languages such as PowerShell. It is designed so that infrastructure update management and integration with other services such as Azure Monitor, Azure Arc for hybrid solutions, and third-party options is a smooth process.

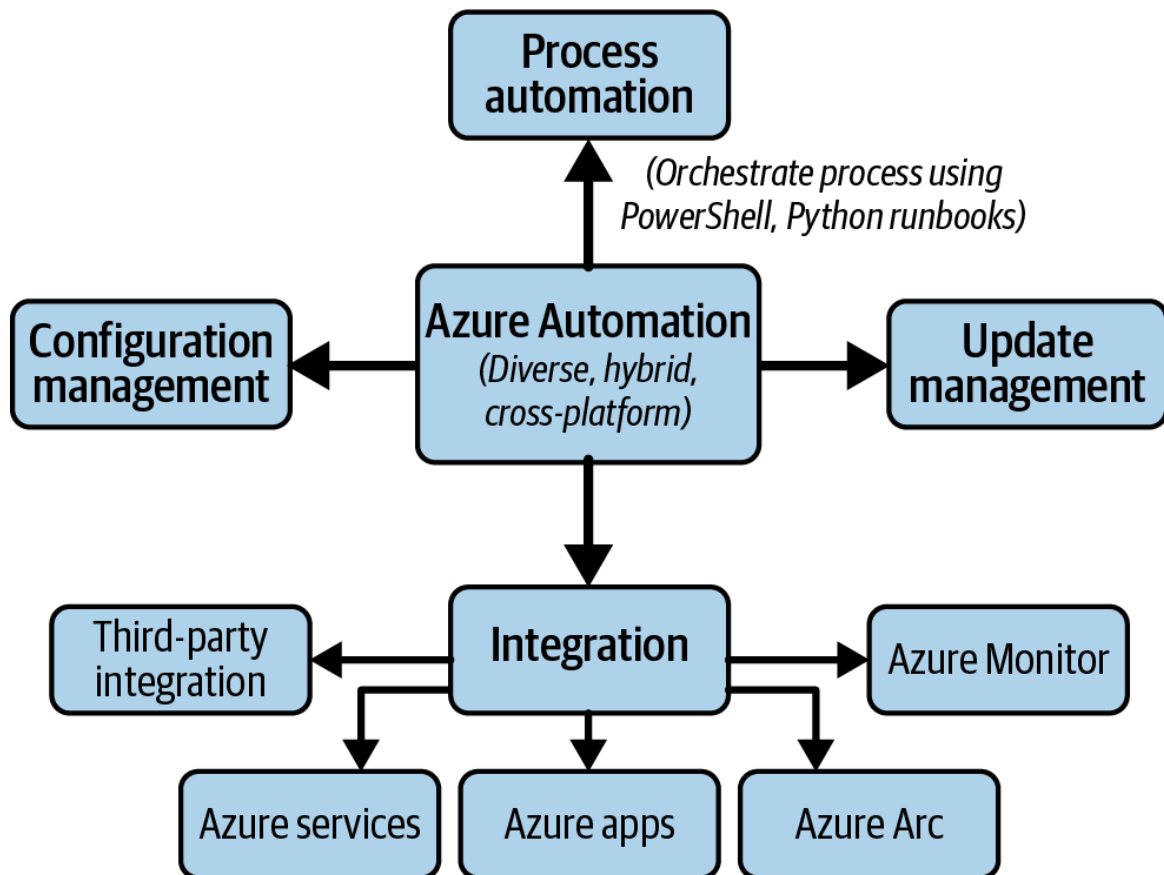


Figure 12-3. Overview of Azure Automation

Benefits of using Azure Automation include:

Automation of repetitive tasks

Allows developers to automate tasks such as provisioning VMs, managing backups, and monitoring system health.

Centralized management

Provides a single platform for managing tasks across Azure and on-premises environments. It simplifies management and reduces the need for multiple tools.

Scalability

Using automation, you can implement autoscaling configuration to enable developers, DevOps, or cloud engineers to quickly add or remove resources.

Using a centralized platform for organizing tasks, developers can reduce the time and effort required to perform routine tasks, freeing up time for more strategic work.

Day-to-day use cases for Azure Automation include:

Provisioning and managing virtual machines

Automate the creation and management of VMs, including provisioning, scaling, and patching.

Managing backups and disaster recovery

Minimize the manual process of performing backups and disaster recovery tasks, such as creating recovery plans and failover processes.

Monitoring system health

Monitor system health and automate responses to alerts such as restarting services or sending notifications.

Automating DevOps

Integrate Azure Automation with Azure DevOps to automate release pipelines, manage IaC, and automate testing and deployment.

Azure Automation is a powerful tool for simplifying the management and automation of tasks in Azure and on-premises environments. Its benefits include automation of repetitive tasks, centralized management, integration with other Azure services, and scalability.

Azure Policy for Compliance and Policy Management

Organizations have management strategies for compliance with regulations and governance policies. Policies are rules that enforce your standards, security, and compliance requirements. Policy implementation in Azure enables you to create, assign, and manage resource policies.

Let's review some of the critical uses of Azure Policy:

Compliance

Ensure that your resources comply with industry regulations and your organization's standards and policies.

Security

Enforce security measures such as access control, encryption, and network security to help protect your resources from unauthorized access or attack.

Governance

Manage and control your resources to prevent unwanted changes and maintain a consistent configuration across your environment.

When setting up policies in Azure, you need to define them, set their assignments, and evaluate them:

Definition

This JSON document defines a policy's rules, scope, and effects. A policy definition includes conditions that must be met for resources to comply with the policy.

Assignment

This assigns a policy definition to a specific scope, such as a resource group or subscription.

Evaluation

This step includes reviewing resources against the assigned policy definition to determine if they comply.

Figure 12-4 shows that an initiative can assign different policies to an Azure subscription or a resource group. One recommended best practice for implementing Azure policies is to ensure you start with a small set to implement. Then you can gradually add new approaches.

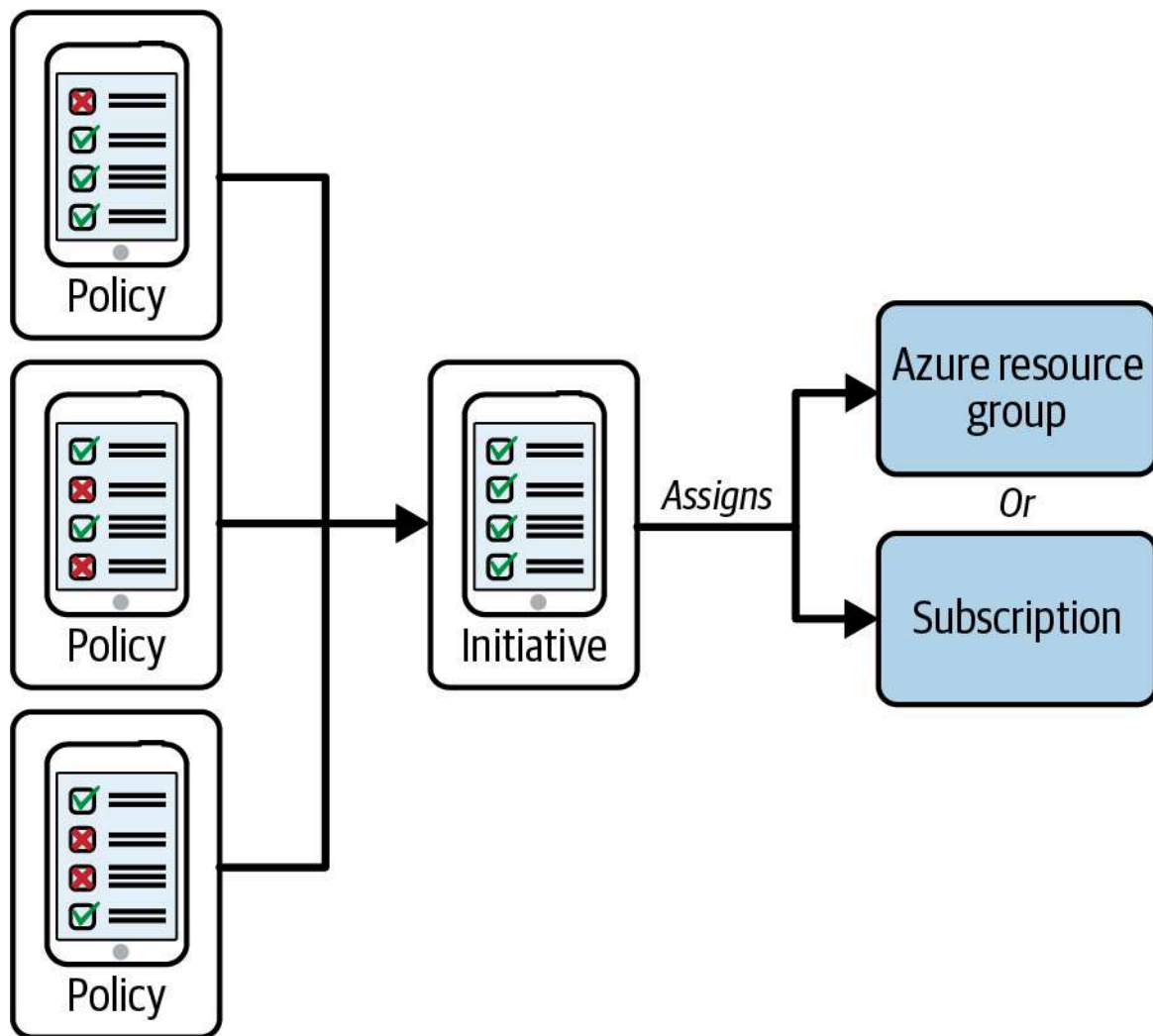


Figure 12-4. Azure Policy initiative assigns policies to an Azure resource group or subscription

Start with creating built-in policies; you can customize them to meet your specific needs. It is also good to group related policies as one single unit for structure. If you want to avoid implementing the procedures directly into production, make it a practice to test and verify these policies in a nonproduction environment.

Overall, Azure Policy is a powerful tool for managing and enforcing your Azure resources' compliance, security, and governance policies to maintain control and visibility in your environment.

FinOps and Cost Management in Azure

FinOps (financial operations) is the practice of managing the financial aspects of cloud computing to optimize costs while maintaining the desired level of performance and functionality. This is particularly important in cloud computing environments such as Azure, where resources can be provisioned and deprovisioned dynamically, making it easy to scale up or down as needed. FinOps aims to help organizations get the most out of their cloud investments while staying within their budget constraints.

Azure Cost Management Tools

Azure provides tools and services to help organizations manage their cloud costs and optimize their spending. One such tool is the Azure Cost Management service, a web-based tool that provides Azure resources cost analysis, budgeting, and optimization features.

Here are some of the benefits of using the Azure Cost Management service and tools:

Cost visibility and transparency

Provides a detailed view of an organization's Azure usage and associated costs. This visibility and transparency enables organizations to identify areas to optimize their spending and reduce costs.

Cost optimization

Provides cost optimization features that enable organizations to identify cost-saving opportunities, such as turning off idle or underutilized resources, resizing or modifying resources, and using reserved instances.

Budget management

Enables organizations to set budgets for their Azure resources and monitor their spending against those budgets.

Sound management helps organizations stay within their budget constraints and avoid overspending.

Cost forecasting

Estimates future Azure costs based on historical usage patterns. This allows organizations to plan their future spending and avoid unexpected expenses.

Integration with other Azure services

Integrates well with Azure services such as Azure Advisor and Azure Monitor, to comprehensively view an organization's Azure resources and associated costs. This integration enables organizations to take a holistic cost management and optimization approach.

Azure Cost Management is a powerful tool for managing cloud costs in Azure. It provides organizations with cost visibility, optimization features, budget management, forecasting, and integration with other Azure services.

Best Practices for Azure Cost Management

Particularly at the beginning of your cloud transformation journey, it's essential to have some guidance as you consider which tools and services to implement while balancing your budget.

Following are recommendations and best practices regarding FinOps in Azure for saving money and optimizing cloud spending:

Understand usage patterns

Understand how your organization uses Azure resources and where the costs come from to identify areas where cost savings can be made.

Use Azure Cost Management

Provide a comprehensive view of your Azure resources and costs to identify cost-saving opportunities, set budgets, and monitor spending.

Use reserved instances

Manage your costs on Azure using reserved instances. You have the option to choose terms for one to three years, which allows you to save money compared to on-demand pricing models.

Configure resources for autoscaling

Automatically adjust the number of resources you're using based on demand to optimize resource usage and reduce costs.

Utilize the Azure Advisor

Optimize your Azure resources for performance, security, and cost.

Monitor and optimize storage costs

Minimize storage costs by monitoring and optimizing your storage usage.

Develop with serverless computing

Run code without managing servers using serverless technologies such as Azure Functions, serverless databases, and containers like Azure Container Apps. This can be a cost-effective way to run applications that don't require a dedicated server.

Use Azure Hybrid Benefit

Use your existing Windows Server licenses with Azure VMs, saving up to 40% on the cost of running VMs.

Track and monitor Azure usage

Periodically review your Azure use and costs to get the most out of your investment.

By following these recommended best practices for cost management, organizations can save money and optimize cloud spending in Azure. Using the right approach, cloud computing can be a cost-effective way to run applications and services that meet your business needs.

Cost Management Optimization for Azure

Cost optimization is a critical reason for managing and controlling costs on Azure. The pay-as-you-go model in Azure helps users and businesses pay only for the resources they use. However, it is easy to overspend on cloud resources without proper cost management.

By optimizing their cloud spend, businesses can reduce unnecessary expenses, get the most value from their investment, and help allocate resources effectively. Since Azure offers a wide range of resources, including VMs, storage, and databases, by managing their costs, businesses can ensure they use the right resources for their needs. For example, cost optimization and management can help avoid underprovisioning or overprovisioning resources, which leads to performance issues or unnecessary expenses.

Cost management is also critical for effective budget planning. Azure gives businesses a clear understanding of their cloud spend, enabling them to allocate resources and plan for future growth accordingly. By clearly understanding their cloud spend, companies can avoid unexpected costs and ensure they invest in the resources needed to grow their business.

Following are cost management and optimization tools Azure provides to its users and enterprises:

Azure Cost Management

A web-based tool that provides cost analysis, budgeting, and optimization features for Azure resources. It enables enterprises to get a detailed view of their Azure usage and associated costs, identify areas where they can optimize their spending, set budgets, and monitor expenditures against those budgets.

Azure Advisor

A tool that recommends optimizing your Azure resources for performance, security, and cost. It helps enterprises identify opportunities to improve resource usage and reduce costs.

Azure Reservations

Azure Reservations enable enterprises to prepay for VMs, storage, and other supported resources for one or three years.

Azure Hybrid Benefit

Hybrid Benefit enables enterprises to use their existing Windows Server licenses with Azure VMs, saving up to 40% on the cost of running those virtual machines.

Azure Spot Virtual Machines

Spot VMs enable enterprises to use spare Azure capacity at a significantly lower price than on-demand pricing.

Azure Budgets

Budgets enables enterprises to set budgets for Azure resources and monitor spending against those budgets. It helps enterprises stay within their budget constraints and avoid overspending.

Azure Monitor

Monitor provides a comprehensive view of an enterprise's Azure resources and associated costs. It enables enterprises to monitor the health and performance of their Azure resources and identify opportunities to optimize resource usage and reduce costs.

Your organization can access your Azure costs using these cost management tools and services. The cost analysis, billing notifications, and other cost-tracking features will help you optimize resource usage and reduce costs using scaling capabilities and features within Azure.

The Evolution of Cloud Management and Governance

The evolution and future of cloud management and governance will likely be characterized by increased automation, standardization, and integration with other IT management and security tools. These trends will likely shape the future of cloud governance.

Multi-cloud management

As organizations adopt multiple cloud providers and services, there will be a growing need for unified tools and platforms to manage and govern these complex environments.

AI and ML

As cloud environments become more complex and dynamic, AI and ML will play a more significant role in automating management and governance tasks. These technologies can help organizations identify and remediate security risks, optimize resource usage, and improve performance.

Cloud-native security

Cloud-native security solutions will become increasingly important as organizations seek to secure their cloud environments from internal and external threats. These solutions must integrate seamlessly with cloud management and governance tools to provide comprehensive security across the entire cloud stack.

Modern cloud and DevOps integration

As organizations continue to embrace DevOps practices, there will be a growing need for cloud management and governance tools that integrate with DevOps tools and workflows. This integration will help organizations manage and govern their cloud environments more efficiently and with greater agility.

Governance, standardization, and compliance

As cloud environments become more regulated, there will be a growing need for tools and frameworks that enable organizations to standardize and enforce compliance across their cloud environments. This will include standardization of security policies, access controls, and other governance-related activities.

Overall, how we work with cloud infrastructure management and its governance going forward will be driven by a need for greater automation, standardization, and integration with other IT management and security tools.

As cloud infrastructure environments become more complex and dynamic, these trends will become increasingly important for organizations looking to maximize the benefits of cloud computing while minimizing risk and cost.

Learn By Doing (Try It!)

The following are recommended tutorials from Microsoft's official documentation:

- [Create and manage policies to enforce compliance](#)
- [Microsoft Azure Fundamentals: Describe Azure management and governance](#)
- [How to use Azure Resource Manager \(ARM\) deployment templates with Azure CLI](#)
- [Automate Azure tasks using scripts with PowerShell](#)
- [Microsoft Learn: Deploy and manage resources in Azure by using JSON ARM templates](#)
- [Tutorial: Create and manage budgets](#)
- [Quickstart: Define and assign a blueprint in the portal](#)
- [Configure Start/Stop VMs during off-hours](#)

Summary

This chapter discussed the services and tools available in Microsoft Azure for cloud management and governance:

- Azure Resource Manager (ARM) enables efficient resource deployment.
- Azure Policy and Blueprints help enforce compliance and governance policies.
- Azure Automation allows for the automation of repetitive tasks.
- Azure Cost Management and Billing help optimize spending.

- Azure Security Center provides advanced threat protection across Azure workloads.

Using these services, organizations can ensure efficient and secure cloud resource management while enforcing compliance and governance policies, ultimately leading to optimized spending and improved productivity.

This chapter also covered considerations for the future of cloud infrastructure management and governance. This space will continue to evolve along with cloud-native and modern technologies.

Check Your Knowledge

1. What is Azure Monitor and how does it help with cloud management?
2. What are some common scenarios in which Azure Automation can be used?
3. How does Azure Resource Manager help with resource deployment and management?
4. What are Azure Blueprints and how do they help with cloud management?
5. What are the best practices for optimizing cloud costs in Azure?

For the answers to these questions, check the [Appendix](#).

Recommended Learning Resources

“Administer Infrastructure Resources in Azure.” Microsoft Learn, <https://oreil.ly/QnIr1>.

“Azure Automation Documentation.” Microsoft Learn, <https://oreil.ly/2xzys>.

“How to Use Azure Resource Manager (ARM) Deployment Template with Azure CLI.” Microsoft Learn, October 12, 2023, <https://oreil.ly/sgXYR>.

“Implement Resource Management Security in Azure.” Microsoft Learn, [*https://oreil.ly/U521M*](https://oreil.ly/U521M).

“Microsoft Azure Fundamentals: Describe Azure Management and Governance.” Microsoft Learn, [*https://oreil.ly/UkdvL*](https://oreil.ly/UkdvL).

Microsoft Developer. “An Overview of Azure Blueprints | Azure Friday.” YouTube video, February 8, 2019, [*https://oreil.ly/Om4pf*](https://oreil.ly/Om4pf).

“Quickstart: Define and Assign a Blueprint in the Portal.” Microsoft Learn, [*https://oreil.ly/jY5Ze*](https://oreil.ly/jY5Ze).

Chapter 13. Cloud Migration, Hybrid, and Multi-Cloud Solutions in Azure

We must never forget our fundamental goal for modernizing and innovating any workload. That is to solve a problem while doing our best not to introduce a new one. This is what makes us responsible technologists. Migrating to the cloud, whether a single cloud, hybrid, or multi-cloud, does not guarantee our success. It does not eliminate the possibility of increasing complexities and causing more damage than the problem we are trying to solve. Our job does not end in migrating to the cloud. Our real journey begins in learning how to adopt, architect, and evolve cloud solutions in the most secure, reliable, streamlined, and sustainable way, so we can genuinely improve people's lives using technology. This is the ultimate test of our knowledge.

—Marilag Svennevig, Chief Solution Architect and
Founder at Dewise, Azure MVP, Cofounder of ULAP.org

Cloud Adoption and Modernization

Cloud adoption is a critical strategic decision for organizations seeking to reduce costs, minimize risks, and achieve scalability in their database capabilities. The extent of cloud adoption may vary depending on an organization's business needs and requirements.

Although migrating to the cloud can be challenging, it can provide numerous long-term benefits for your business. Cloud management platforms such as Microsoft Azure can monitor and optimize your cloud costs. They enable you to accurately track your expenses, identify potential

cost-saving opportunities, and optimize your cloud environment, enhancing efficiency.

As organizations increasingly rely on technology to run their operations, the cloud is essential for their teams, IT architects, and developers to build and deploy applications quickly and efficiently. Organizations can benefit from significant improvements in cost, agility, scalability, and security by adopting and modernizing their IT infrastructure with cloud technologies.

Figure 13-1 demonstrates the sequence of the cloud adoption model. The components are described following the figure.

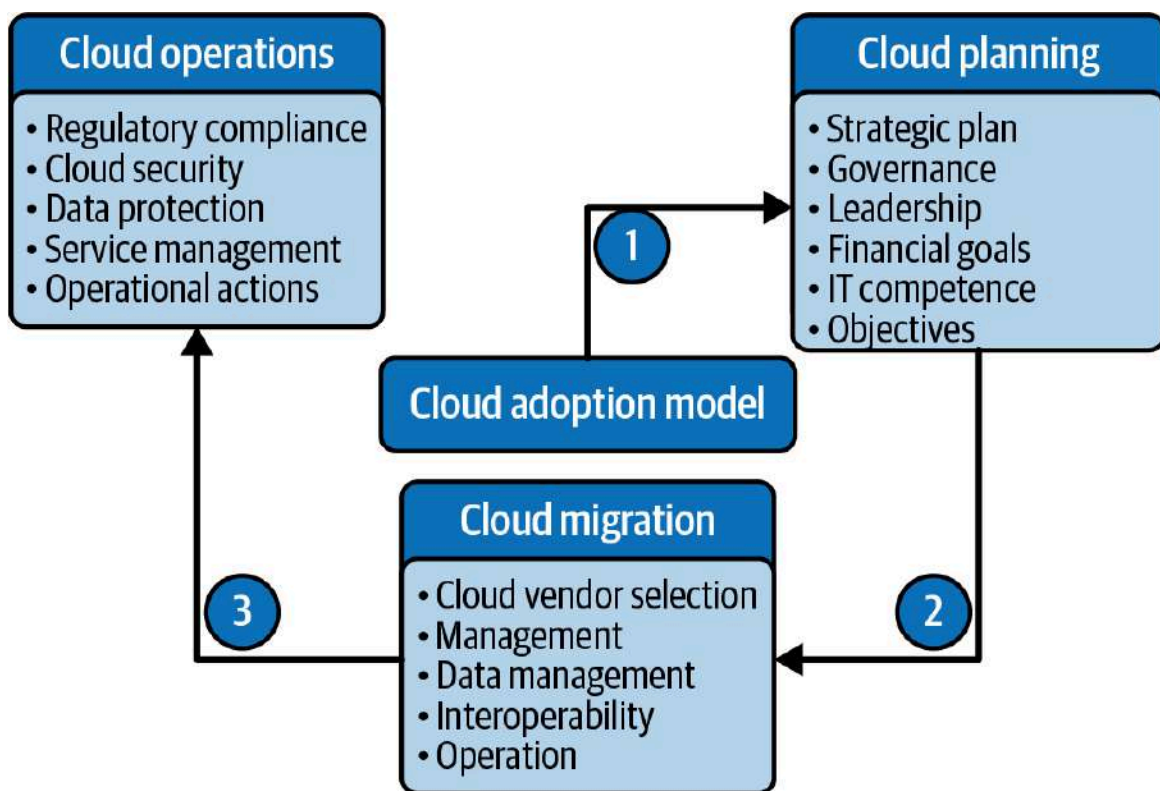


Figure 13-1. Cloud adoption model processes and steps

Cloud adoption allows organizations to move away from traditional on-premises infrastructure and shift to a more flexible and dynamic cloud-based infrastructure. With cloud computing, IT architects and developers can use on-demand, easily scalable resources to respond to changing business needs more quickly and efficiently.

1. Cloud planning

Planning is the first step in any cloud transformation and adoption process. It includes identifying critical factors for governance strategies, such as the leadership team that leads the cloud innovation, and how to gather technical competencies, financial goals, and more.

2. Cloud migration

This subsequent phase is designing and implementing the cloud migration. It includes selecting the cloud vendor that suits the plan in the first step and strategies for migrating existing workloads to the cloud. It also includes considering management operations in terms of interoperability, management of data, processes, monitoring, and other essential migration steps.

3. Cloud operations

Once you've migrated to the cloud, the infrastructure and workloads hosted on the cloud must be monitored, maintained, and secured for reliability. Taking care of operations hosted on the cloud means managing practical factors such as compliance, security, data protection, operations agility, and management of services.

While this may sound complex, cloud computing helps increase businesses' agility by enabling faster time-to-market for new applications and services and giving organizations a competitive edge in today's rapidly evolving market.

In addition to agility, cloud adoption can also bring significant cost savings. Cloud computing eradicates the need for expensive hardware, software, and infrastructure maintenance, freeing up IT budgets for other strategic investments. By moving to a cloud-based infrastructure, organizations can pay for only the resources they use, making it easier to scale up or down as needed while avoiding unnecessary expenses.

Modernization of Legacy Applications and Traditional

Infrastructure

Modernizing legacy applications and infrastructure is another critical advantage of cloud adoption. By migrating to the cloud, organizations can use new cloud-native services that are purpose-built for modern application development, such as containers, serverless computing, and microservices. These services can help IT architects and developers build and deploy applications more quickly and easily, reducing development and deployment times and improving application performance and scalability.

Cloud adoption also improves security. Cloud providers typically offer better security measures and protocols than most organizations can implement. This includes security features such as data encryption, identity and access management, and intrusion detection and prevention, ensuring that data and applications are safe and secure.

MY CLOUD MIGRATION JOURNEY AS DEVELOPER AND ARCHITECT

As many organizations and users depend on technology to run their operations and perform their tasks, the cloud is essential for IT architects and engineers to build and deploy applications quickly and efficiently. However, many organizations need help migrating their legacy applications to the cloud, particularly when they need a clear cloud adoption strategy and an understanding of the benefits of cloud technology.

As someone who has worked on a cloud migration journey, I can attest to the importance of developing a comprehensive cloud adoption strategy and understanding the benefits of cloud technologies. In my previous role, I was responsible for migrating a 15-year-old legacy application system on .NET hosted on premises to the cloud on Azure. Despite trying different options, such as lift-and-shift, rearchitect, and modernization, the project did not make it to production.

After careful consideration, the solution that worked was creating a new system. By removing the barriers that caused a lack of a clear cloud adoption strategy and knowledge of cloud benefits among IT teams, the organization was able to contribute to the project's success. This experience highlighted the importance of developing a concrete cloud adoption strategy and building a culture of cloud adoption to ensure that cloud technologies can deliver their full potential.

While cloud adoption and modernization offer significant benefits to organizations, they require a comprehensive strategy, a clear understanding of cloud benefits, and a culture of adoption to realize their full potential.

Staying up-to-date with the latest cloud technologies and best practices is essential to help organizations make informed decisions about their cloud adoption strategy and avoid common pitfalls.

Digital Estate and the Prerequisites of Cloud Migration

Cloud migration can challenge organizations and IT teams. Understanding available options can help organizations and IT teams execute their cloud migration. Once those options are understood, it is also essential to make wise decisions and find distinct factors to migrate successfully.

In today's digital landscape, every company has a digital estate, a collection of tangible owned assets that power business processes and supporting operations. Like a physical estate, a digital estate consists of various elements, such as VMs, servers, applications, and data.

Understanding the importance of a digital estate is crucial when planning and executing digital transformation initiatives. During transformation journeys, cloud strategy teams use the digital estate to map business outcomes to release plans and technical efforts.

The measurement of a digital estate varies depending on the desired business outcomes, for example, infrastructure migrations focus on VMs, servers, and workloads to optimize costs, operational processes, agility, and other aspects of operations. On the other hand, application innovation emphasizes applications, APIs, and transactional data that support customers, with less focus on VMs and network services. Therefore, cloud-enabled data innovation efforts focus more on silos of data across the organization.

Operational stability is also a top priority for businesses. This requires measuring the positive or negative impact of the digital estate on stable operations, including business continuity, disaster recovery, and workload and asset reliability. Understanding and measuring the digital estate is crucial for companies to optimize operations, improve customer experiences, launch new products or services, and achieve operational stability.

Identifying your organization's digital estate

Before an organization can move its digital assets to the cloud, it must understand its digital estate, including its size, scope, and characteristics.

Understanding it is critical for optimizing costs, improving security and compliance, supporting digital transformation, and planning and executing cloud migration.

By knowing what assets in the infrastructure are part of the digital estate, organizations can identify inefficiencies and redundancies in their IT infrastructure. This can lead to cost savings by reducing the number of assets used and optimizing resource allocation.

Additionally, identifying an organization's digital estate can help improve security and compliance. Knowing what assets are part of the digital estate makes it easier to manage access controls, monitor for vulnerabilities, and ensure compliance with regulations and standards.

The digital estate also serves as a foundation for digital transformation goals. By knowing what assets are part of the digital estate, organizations can identify what needs to be modernized, migrated to the cloud, or replaced.

Here are steps that organizations typically take to accomplish this:

1. Conduct an IT asset inventory: The first step is to create a comprehensive list of all the IT assets that the organization currently owns, such as applications, databases, servers, and other hardware and software components. This may involve working with various teams and departments.
2. Categorize and assess IT assets: Once the inventory is complete, the organization should categorize and consider each asset based on its characteristics and business importance. For example, assets may be organized by application type, data sensitivity, compliance requirements, or other factors relevant to the organization's goals and objectives.
3. Evaluate IT assets for cloud suitability: The organization should then evaluate each asset to determine its suitability for migration to the cloud. This may involve assessing performance, security, compliance,

cost, and other considerations relevant to the organization's IT strategy and goals.

4. Plan the migration: Based on the assessment results, the organization should develop a migration plan outlining how each asset is being migrated to the cloud. This step may involve selecting the right cloud platform, determining the migration approach (lift-and-shift, replatforming, or rearchitecting), and defining the timeline and resources needed for the migration.
5. Test and validate: Before finalizing the migration, the organization should test and validate to ensure that the migrated assets are functioning correctly and meeting the desired performance, security, and compliance requirements. This may involve tools and techniques like load testing, vulnerability scanning, and compliance audits.

Cloud Rationalization

Cloud rationalization refers to an organization's process of assessing its existing IT assets, such as applications, databases, or servers, and deciding which ones should be migrated to the cloud and how they should be migrated. This process aims to identify the most effective and efficient approach for each asset based on its specific characteristics and business requirements.

Rationalization implies that the process involves making logical and reasoned decisions about which assets to migrate and how to migrate them. This can include evaluating each asset's performance, security, cost, and compliance implications in the context of the organization's overall IT strategy and goals.

Overall, cloud rationalization aims to help organizations make informed decisions about leveraging the benefits of cloud computing while minimizing risks and maximizing the value of their IT investments.

The five Rs of rationalization

The five Rs of rationalization are available application cloud migration strategies for organizations and IT teams looking to migrate and use Azure.

Rehost or the lift-and-shift

This strategy involves migrating existing applications on premises to the cloud, with minimal changes to the source code, applications, databases, and other cloud resources. This approach is often used for stable applications with minimal dependencies. The most common reason organizations consider this first is to quickly achieve a return on investment (ROI).

Refactor

Refactoring is restructuring existing code to improve quality and functionality without changing external behavior. In modern application development for the cloud, refactoring is often done to enable an application to take advantage of new business opportunities. To achieve this, developers may consider deploying their applications on a platform as a service (PaaS) model, which can help reduce operational costs associated with application hosting. It is sometimes a wise decision for many organizations to use PaaS services because it enables them to efficiently utilize the cloud by providing scalability and a consumption-based pricing model.

Rearchitect

Rearchitecting refers to modifying and extending the existing codebase of the applications to be cloud-native. For example, some legacy applications may not have been designed with cloud computing in mind and may not be easily moved to cloud-based services. To make these applications suitable for the cloud, they may require

rearchitecting. This process involves reworking the application's architecture to optimize it for cloud-based deployment.

Rebuild

In certain situations, the technological disparity between an application and the cloud platform can be insurmountable, making investing more in the application unfeasible. This challenge is particularly prevalent for legacy applications that were once sufficient for business needs but are now incompatible with current business processes. One solution to this problem is to rebuild, which means creating a new codebase that aligns with cloud-native practices. During this process, organizations can also leverage the opportunity to incorporate DevOps methodologies into their application development and deployment for cloud platforms. By adopting agile DevOps techniques with cloud development, businesses can accelerate their application delivery cycle; improve reliability, security, and scalability; and streamline their cloud deployment process. Overall, rebuilding the application or creating a new codebase that aligns with cloud-native practices can help businesses modernize their legacy applications and overcome the technological disparity between their applications and the cloud. Additionally, incorporating DevOps methodologies can further enhance the efficiency and effectiveness of application development and deployment.

Replace or retire

In some instances, replacing or retiring an application may be necessary. If a custom application is proving challenging to support, an available software as a service (SaaS) option could provide the required functionality. Alternatively,

another product within the organization may have modules or features that can replace the application's functionality.

The five Rs of rationalization for Azure provide different options for migrating applications to the cloud. Understanding the available options can help organizations and IT teams plan and execute a successful migration to Azure. Organizations and their IT teams need to learn about these different strategies to make informed decisions about their cloud migration strategy and ensure a successful migration to Azure.

Finally, it is important to note that various methods are available for modernizing an application for the cloud. However, it is crucial to assess each application individually to determine the most appropriate modernization strategy. A one-size-fits-all approach often leads to suboptimal results. Therefore, it is recommended to analyze each application independently to determine the most effective application modernization strategy.

Cloud Adoption and Migration Anti-Patterns

There are a few **cloud migration anti-patterns** that IT organizations, IT architects, and software engineers should be aware of when using the **Cloud Adoption Framework (CAF) for Azure**:

Lift-and-shift approach

This anti-pattern involves simply moving existing on-premises applications to the cloud without redesigning or optimizing them for the cloud environment. While this approach may look like a quick fix, it can result in deficient performance, higher costs, and missed opportunities for cloud-native features and services.

Big Bang migration

This anti-pattern involves migrating all applications and workloads to the cloud simultaneously. This can result in increased complexity, higher risks, and longer downtime. A phased approach is recommended instead.

Lack of security and compliance considerations

This anti-pattern involves not considering security and compliance requirements during migration. This can result in security breaches, compliance violations, and reputational damage.

Overprovisioning resources

Right-sizing the provisioning (neither over nor under) involves providing more resources in the cloud than are needed, resulting in higher costs and underutilized resources. Right-sizing and optimization of resources are recommended instead.

Neglecting to optimize for cloud-native features

This anti-pattern involves not taking advantage of cloud-native features and services, resulting in missed opportunities for cost savings, performance improvements, and scalability.

To avoid these anti-patterns, IT organizations preparing for cloud transformation and adoption should follow the best practices and guidance provided by the **Cloud Adoption Framework (CAF)** for Azure. This includes conducting a thorough assessment of the current environment, developing a detailed migration plan, considering security and compliance requirements, optimizing resources, and taking advantage of cloud-native features and services.

The Five Pillars of a Well-Architected Framework for Azure

The Well-Architected Framework (WAF) for Microsoft Azure let's look at the pillars of this framework that help organizations prepare for their cloud migration projects.

Figure 13-2 shows the pillars: operational excellence, security, reliability, performance efficiency, and cost optimization.

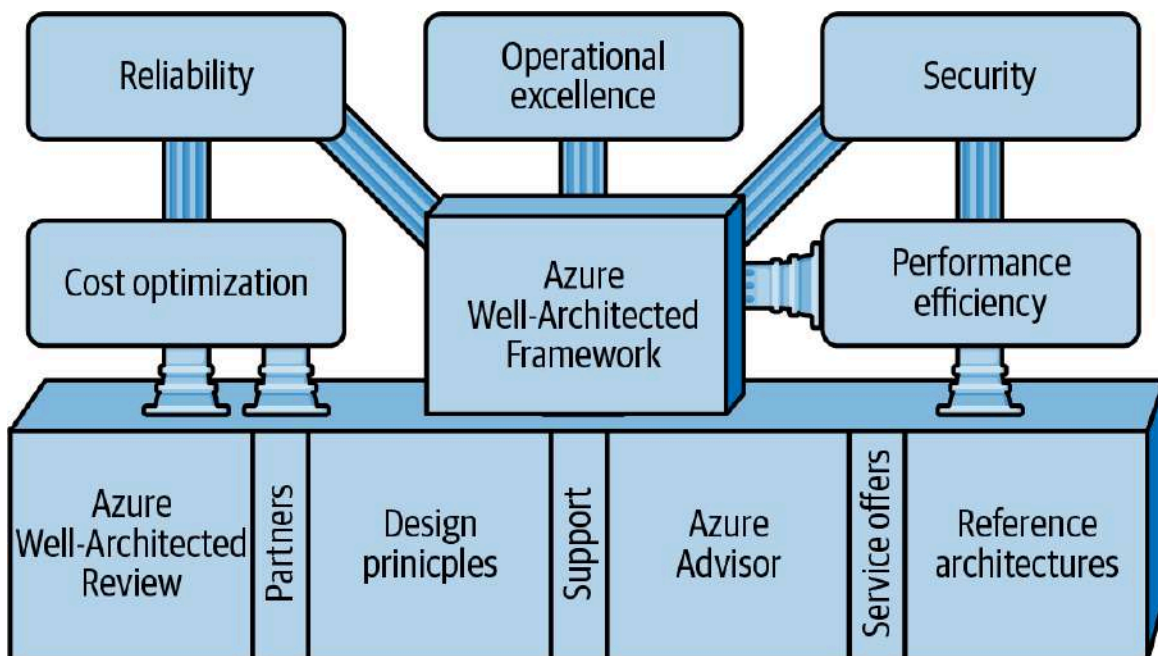


Figure 13-2. Critical pillars of the Well-Architected Framework for Azure

Let's take a closer look at each pillar.

Pillar #1: Operational Excellence

Operational excellence ensures that systems are reliable, scalable, and easy to manage. It includes monitoring, automation, resource optimization, and disaster recovery. Moreover, this pillar also aims to help organizations achieve operational excellence by following best practices for cloud infrastructure management.

Here are some key concepts for operational excellence:

Operations monitoring

Involves monitoring systems to ensure they function as intended and identifying potential issues before they become problems.

Change management

Concerns managing system changes, including software updates, and infrastructure changes, to ensure they are implemented smoothly and without disrupting system availability.

Disaster recovery

Involves planning to recover from disasters, such as natural disasters or cyberattacks, and ensuring that systems can be quickly restored to their previous state.

To achieve operational excellence, you should also consider how automation and optimization can help your business.

Automation of processes

The main purpose of automation is to develop solutions and tools to reduce the manual processes done by humans. An automated approach is better suited to functions and tasks that can be tedious, repetitive, and time-consuming. When we know we expect predictable results, we can develop an error-free automation tool.

Automation in this context can mean automating deployment processes, scaling the servers, and autoscaling based on workload demands to reduce the risk of human error and increase efficiency. If possible, automate and remove manual processes. There are different technologies and tools available to make it possible; in Azure, serverless solutions like Azure Functions and Azure Container Apps enable integration with applications with automation capabilities.

Examples of automation in Azure include:

Automation of infrastructure management

Implementing automation tools, such as infrastructure as code (IaC) tools like Terraform, Azure Bicep, Chef, Puppet, Azure PowerShell, Azure CLI, and ARM templates can help manage infrastructure and reduce the risk of human error.

Intelligent monitoring and alerting

Automating your monitoring systems using Azure Monitor, Application Insights, and Microsoft Defender for Cloud for security can help cloud administrators and organizations monitor server performance and security and send alerts when issues are detected.

Disaster recovery planning

Developing a disaster recovery plan with automation helps you configure and control how your systems get restored or replicated during a disaster.

In addition, sturdy **release engineering** for application deployments, test automation, SRE, and DevOps are good practices to consider.

Finally, check out this standard operational excellence pattern recommendation and **checklist for monitoring DevOps**.

Cloud resource optimization

It's important to optimize cloud resources, such as storage and computing power, to reduce costs and improve performance.

Overall, the operational excellence pillar is essential for organizations that want to achieve operational excellence in their cloud infrastructure management. Organizations can build and operate reliable and scalable systems on Azure by following best practices for monitoring, automation, disaster recovery, and resource optimization.

THE MEANING OF IDEMPOTENCE IN CLOUD AUTOMATION

Reliable operations are automated and idempotent—that is, repeatable to produce the same results. Idempotence is a property of automation in which an action can be repeated multiple times without causing unintended consequences or changing the outcome of the previous step. In other words, an idempotent action creates the same result regardless of how often it is executed.

In cloud automation, idempotence is critical because automation systems often repeat actions multiple times. Suppose an automation system creates a VM in the cloud; it may need to repeat that action numerous times to create multiple instances of that VM.

If the action is not idempotent, repeating it multiple times could lead to unintended consequences or errors. If the step is to create a new VM, repeating the action numerous times could create multiple VMs with the same name or configuration, causing confusion and potential errors in the system.

To ensure idempotence in cloud automation, automation scripts or tools should be designed to check for the existence of the resource or state of the system before executing the action. If the resource or a certain condition already exists, then the activity should be skipped or marked as completed. Deploying Azure resources through infrastructure as code (IaC) using Azure Terraform or the Bicep language are two good examples of this.

Pillar #2: Security

Security focuses on ensuring systems are secure and protected against threats. It covers identity and access management, network security, and data protection.

This pillar aims to help organizations design and operate secure systems hosted and provisioned on Azure. It provides best practices and guidelines for implementing security controls to protect against threats, manage identity and access, and comply with regulations and standards.

Fundamental security components of the WAF for Azure include:

Identity and access management (IAM)

Controlling identities, credentials, and permissions for accessing cloud resources. IAM best practices include implementing multi-factor authentication and least privilege access controls, and auditing access events.

Data protection

Implementing encryption, network segmentation, and data backup and recovery to protect sensitive data from unauthorized access, disclosure, and loss.

Network security

Implementing firewalls, access control lists, and network security groups to protect networks from external and internal threats.

Threat detection and response

Implementing security monitoring and incident response capabilities to promptly detect and respond to security incidents.

Regulations and compliance

Complying with regulations, standards, and best practices related to security, such as HIPAA, PCI DSS, and ISO 27001.

The typical implementation also includes using Azure Firewalls to protect web applications from attacks such as SQL injection and cross-site

scripting.

It's also essential to protect data transmitted over public networks. For example you can protect the data in transit between applications over the internet by implementing SSL/TLS encryption.

Furthermore, consider checking the framework's **security design principles** to help build and implement security in your systems.

Pillar #3: Reliability

No system is perfect; failure is inevitable, even in our applications. The reliability pillar encourages organizations, IT architects, and developers to design their architectures with loss in mind, ensuring they can quickly detect and respond to issues and minimize the impact on users. This pillar focuses on your system architecture, applications, and workloads on the cloud being reliable, resilient, and durable. These characteristics are essential in systems that need to be available to users 24/7. Reliability also includes ensuring the foundation of your systems architecture keeps your applications reliable, and making sure you take into account failure management, change management, and performance efficiency.

The reliability pillar encourages designing architectures with failure in mind, implementing automatic scaling, and establishing automated backups and disaster recovery procedures to mitigate the impact of losses.

Figure 13-3 illustrates how cloud resources in Azure, such as Azure VMs and App Services, can be scaled vertically or horizontally for automatic scaling. Azure allows users to configure their workload scaling based on the demands in their environments.

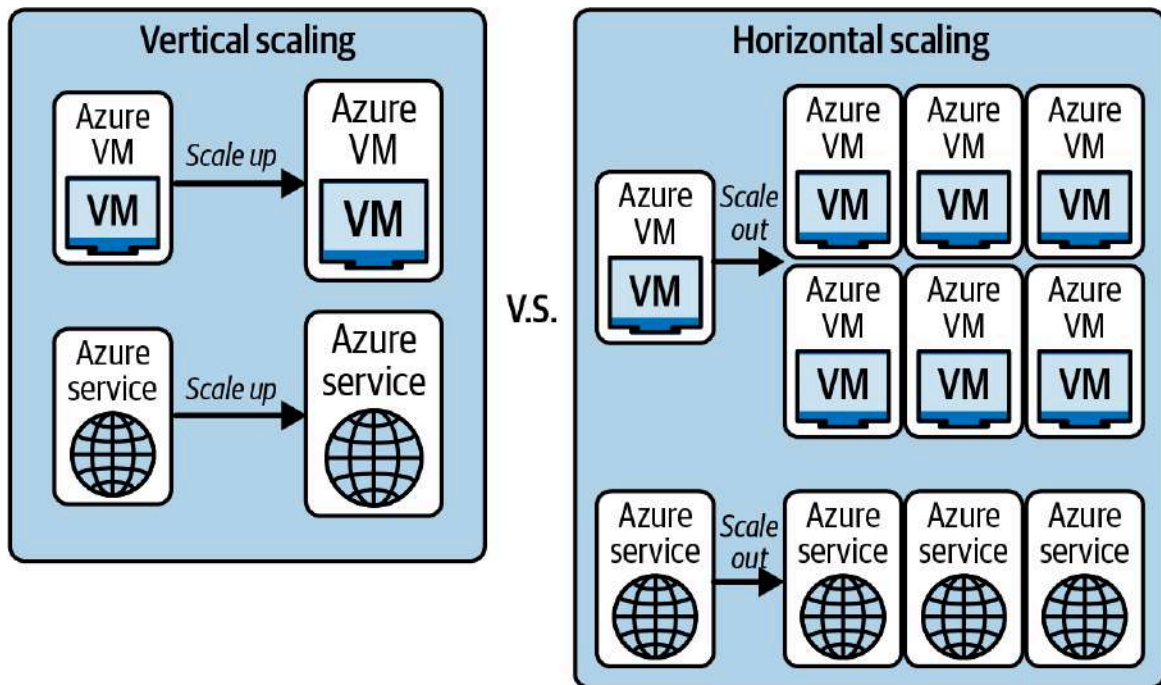


Figure 13-3. How scaling works in Azure, specifically when scaling Azure VMs and App Services

Change management

Change management is another critical aspect of reliability. As workload and feature requirements change over time, architectures must be able to adapt without causing downtime or disruptions. It is a crucial step in any IT environment, including cloud environments such as Azure. Change management controls changes to the IT infrastructure, including hardware, software, and processes.

In Azure, change management is particularly vital because it helps ensure the cloud infrastructure's reliability, availability, and security for supporting business operations. Implementing testing and validation processes, configuration management procedures, and effective monitoring and alerting systems are essential to do this effectively.

An example use case that can be helpful for developers, businesses, and IT projects when considering the concept of reliability is designing a distributed system. When designed well, they can handle component-level failures without impacting users, implement retry operations to solve transient errors, and automatically scale to address changes in workload.

It is also crucial to implement effective monitoring and alerting systems, robust testing and validation processes, and configuration management procedures to ensure that changes to the architecture do not introduce unintended consequences or risks.

Here are recommended best practices for Azure WAF reliability:

- Design a reliable distributed system that can handle failure at the component level without causing downtime for users.
- Develop applications that handle transient errors and retry operations automatically, minimizing impact on the user.
- Implement an automatic scaling configuration to ensure that your system can handle changes in workload without impacting performance or availability; see **Figure 13-3** for an example of vertical scaling versus horizontal scaling.
- Implement automated backups and disaster recovery procedures so your system can recover quickly in the event of a catastrophic failure.
- Perform effective monitoring and alerting to detect and respond to issues before they impact users.
- Create a robust testing and validation process to ensure changes to your system do not introduce unintended consequences or risks.
- Enforce a configuration management process to ensure that changes to your system are tracked and audited, reducing the risk of configuration drift and security issues.

Pillar #4: Performance Efficiency

Performance efficiency focuses on improving efficiency of your workloads while minimizing costs. It ensures that your architecture meets business requirements without incurring unnecessary costs or causing performance bottlenecks.

Here are key ways to accomplish optimal performance and efficiency:

Compute and storage optimization

It is important to carefully consider the performance and cost implications of choosing different computing and storage resources. This includes evaluating instance sizes, storage types, and networking configurations to ensure the selected resources are optimized for their workload.

Designing architectures for elasticity

Elasticity is a critical component of any well-architected system, particularly for expert readers who must design architectures that can handle complex, dynamic workloads. This involves creating systems that can automatically scale in response to changes in demand, ensuring that resources are allocated only as needed to avoid overprovisioning or underprovisioning.

Investigating and monitoring performance issues and bottlenecks

Optimizing performance requires monitoring, investigation, and a deep understanding of how applications function, including identifying errors, poorly performing code, and bottlenecks in different systems. These issues can be effectively uncovered and addressed using application performance-management tools such as Azure Application Insights, Data Dog, or New Relic, which can reveal hidden or obscured issues that may otherwise impact the overall performance of your application, affecting users, developers, and administrators alike.

Strategies for optimizing performance

Performance optimization is a critical aspect of performance efficiency, particularly for expert readers responsible for ensuring their workloads perform optimally. This includes tuning database performance, optimizing network

configurations, and implementing caching and CDNs to improve response times and reduce latency.

Moreover, by focusing on performance efficiency as part of a WAF, organizations can optimize costs, deliver a high-quality experience to their application users, and be assured that their systems are responsive, high-performance, scalable, and elastic.

Pillar #5: Cost Optimization

Cost optimization is the final pillar of the Azure WAF. It drives business. It focuses on helping organizations minimize their cloud infrastructure costs while still meeting their business requirements and delivering value to their users and customers. The goal is to achieve the best possible balance between cost and performance without sacrificing security, reliability, or other vital factors.

Cost optimization includes:

Evaluate cost drivers

It is essential to clearly understand what factors contribute to your infrastructure costs to identify areas for improvement. This includes identifying wasteful or unnecessary spending and analyzing the prices of different resources and services.

Properly provision Azure resources

Many organizations overprovision their resources, leading to unnecessary costs. By right-sizing resources, you can ensure you are only paying for what you need without sacrificing performance. Azure Reservations gives organizations the option to save yearly costs by subscribing to one-to-three-year product plans with Microsoft.

Use cost-effective architectures

Many different architectural approaches can help organizations reduce infrastructure costs, such as using serverless computing, leveraging managed services, and implementing cost-effective storage solutions.

Optimize resource use

Ensure your resources are used efficiently to avoid waste and reduce costs. This can include automating resource provisioning and deprovisioning, implementing effective monitoring and alerting, and using automation to optimize resource use.

The main benefits of focusing on cost optimization is reduced infrastructure costs. Optimizing your resources and taking a cost-effective approach to architecture can reduce your infrastructure costs, freeing up resources for other business areas. This can surge throughout your organization, letting you invest more in other areas, such as marketing, research, development, or hiring.

Another critical benefit of cost optimization is better cost management. When you focus on reducing your infrastructure costs, you better understand what factors drive your spending and can identify improvement areas. This can help you better manage infrastructure costs, track spending, and avoid unexpected expenses. With a better understanding of your costs, you can make better decisions about allocating resources and investing in new initiatives for the cloud.

Benefits of cost optimization of cloud resources

In today's fast-paced business environment, every advantage counts, and focusing on cost optimization can help you stay ahead of the curve. Optimizing costs can provide an organization with a competitive advantage. Businesses can gain a competitive edge over other organizations that overspend on their infrastructure. This can allow you to offer competitive pricing or invest in new initiatives.

Cost optimization can also lead to improved business agility. By reducing costs, you can make your organization more agile and responsive, allowing you to innovate and adapt quickly. It can be imperative in fast-moving industries, where being able to adjust and adapt to changes in the market can be the difference between success and failure. Optimizing your infrastructure costs can create a more flexible, adaptable organization better equipped to handle changing circumstances.

By following the best practices and **design principle guidelines for cost optimization** drafted in the WAF, organizations can improve the overall quality of their systems, reduce costs, and better align their cloud infrastructure with their business goals.

Cloud Adoption Framework for Azure

The Cloud Adoption Framework (CAF) for Azure provides best practices and guidance for organizations to adopt and deploy Azure cloud services. It is a structured approach to cloud adoption that helps organizations to migrate workloads to Azure, deploy infrastructure, and optimize Azure resources.

CAF has seven core components essential for Azure adoption: strategy, plan, ready, adopt, govern, manage, and secure. Each component has a specific set of actions and best practices to be followed to ensure a smooth, successful Azure adoption. **Figure 13-4** shows each of these components as steps within the framework that are continuously improved, repeated, and iterated.

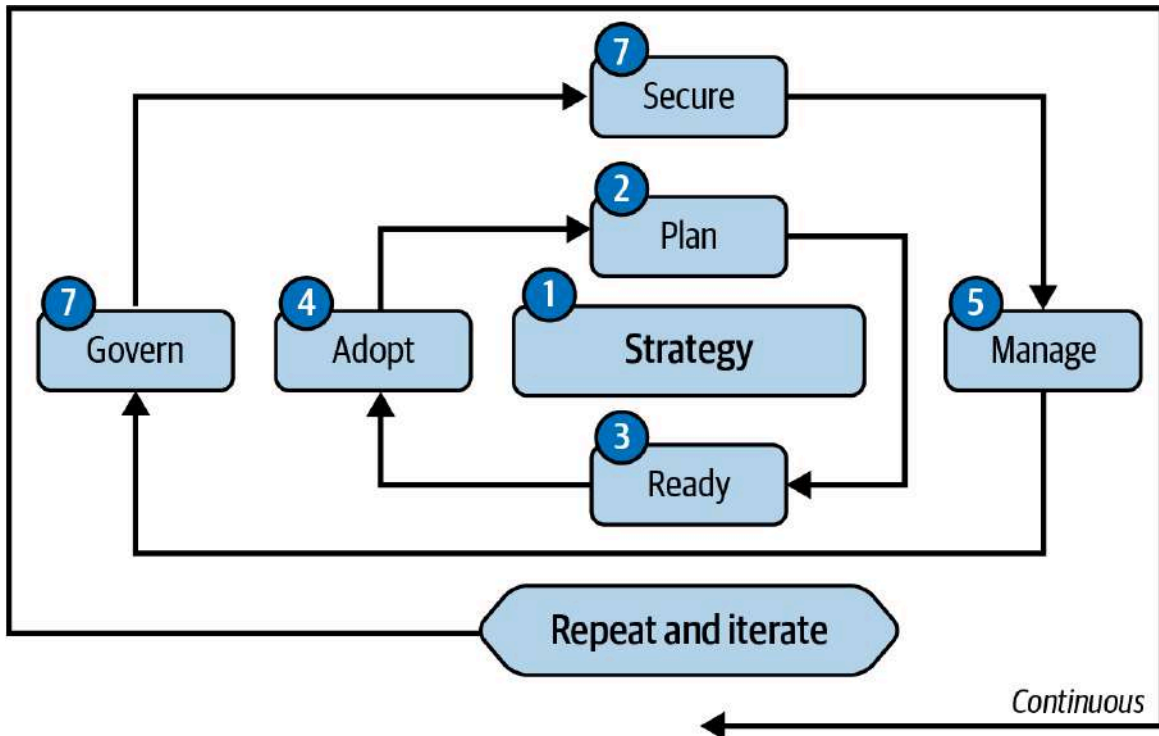


Figure 13-4. The Cloud Adoption Framework for Azure is continuous and iterative

Let's take a closer look at each core component of the CAF:

1. Strategy

This component helps organizations define their cloud adoption strategy and goals. This includes defining the business drivers and objectives, assessing readiness for cloud adoption, and identifying the key stakeholders and decision-makers.

Example: An organization might move to the cloud to reduce infrastructure costs, improve agility, and enhance team collaboration. The organization's cloud adoption strategy might involve migrating its on-premises applications and data to Azure and adopting a DevOps software development and delivery approach.

2. Plan

This component helps organizations plan and prepare for their cloud adoption. This includes defining the migration approach, identifying

the workloads to migrate, and preparing the technical and operational teams for the migration.

Example: An organization might create a detailed migration plan that outlines the migration process for each application, including the dependencies, risks, and timelines. They might also establish a cloud governance model to ensure the migration is controlled and efficient.

3. *Ready*

The CAF's ready component helps organizations prepare their technical and operational teams for cloud adoption. This includes providing training and support to the teams, defining cloud roles and responsibilities, and establishing the necessary tools and processes for cloud adoption

Example: An organization might provide training to its IT staff on Azure services and tools, such as Azure Resource Manager and Azure Security Center. They might also establish a cloud center of excellence (CCoE) to provide guidance and support for the cloud adoption process.

4. *Adopt*

This component involves the actual deployment and adoption of Azure services. This includes migrating the workloads to Azure, deploying the necessary infrastructure, and optimizing resources in Azure.

Example: An organization might use Azure Site Recovery to replicate its on-premises VMs to Azure, enabling it to migrate applications to Azure without downtime. It might also deploy IaC using Azure Resource Manager templates to automate the deployment and configuration of its Azure resources.

5. *Manage*

This component helps organizations manage and optimize their Azure resources, for example, monitoring the performance of Azure

resources, optimizing resource usage, and identifying opportunities for further cost optimization.

6. *Govern*

The govern component helps organizations establish and enforce policies for Azure adoption. This includes defining security and compliance requirements, establishing a cost management framework, and monitoring Azure resources for compliance and security.

Example: An organization might establish policies for data classification and access control, defining who has access to what data in Azure. It might also use Azure Cost Management and Billing to monitor Azure usage and optimize costs, and use Azure Security Center to monitor Azure resources for security threats and vulnerabilities.

Example: An organization might use Azure Monitor to monitor the performance of Azure resources, identify any issues or bottlenecks, and troubleshoot problems. It might also use Azure Advisor to identify optimization opportunities and implement recommendations to optimize its Azure usage and reduce costs.

7. *Secure*

The secure component of CAF helps organizations establish and enforce security controls and compliance requirements for Azure resources. This includes identifying security risks and threats, implementing security controls and best practices, and monitoring Azure resources for security incidents.

Example: An organization might conduct a risk assessment of Azure resources and identify security risks and threats. It might implement security controls such as network security groups and VPNs, to secure Azure resources. It also might use Azure Security Center to monitor Azure resources for security incidents and respond to security alerts promptly.

Overall, these core components provide a structured approach to cloud adoption and deployment in Azure, enabling organizations to achieve their cloud adoption goals in a controlled, efficient, and optimized manner.

Although security is often missed in these phases of cloud adoption, it is critical to include in the process. The CAF's secure component helps organizations ensure the security and compliance of their Azure resources, protecting their data and applications from security threats and vulnerabilities.

Benefits of the Cloud Adoption Framework for Azure

Now that we understand the framework, let's look at the benefits of using this approach to deploy Azure cloud services.

Structured approach

The CAF provides a structured approach to cloud adoption that helps organizations follow a standardized process, reducing the risk of errors and inconsistencies during the adoption process.

Recommended best practices

The CAF provides best practices and guidance for Azure adoption based on Microsoft's experience with Azure adoption across various industries and organizations.

Flexible and customizable

Implementing the CAF provides flexibility, and it is customizable to meet the specific needs of each organization. It provides a set of core components that can be tailored to fit the organization's unique requirements and goals.

Increased efficiency

Following CAF best practices and guidance helps organizations streamline the adoption process. Increasing efficiency signifies reducing the time and effort required to deploy Azure services.

Cost optimization

The CAF helps organizations to optimize costs by guiding resource management, automation, and cost optimization techniques. This helps organizations reduce their overall Azure spend while ensuring their resources are used efficiently.

Improved security

The CAF guides security best practices for Azure adoption, helping organizations ensure that their Azure resources are secure and compliant with industry standards and regulations.

Recommended Approach for Cloud Adoption in Azure

There are several recommended approaches to cloud adoption in Azure based on industry experience and Microsoft's expertise in cloud computing. These recommended CAF approaches in Azure ensure a successful, sustainable cloud adoption journey.

Let's review each of these approaches that help organizations minimize risk, maximize benefits, and continuously improve their cloud adoption process.

The 4 S's: Start Small Smart Steps

Starting small is a recommended approach to cloud adoption in Azure, as it allows organizations to gain experience and confidence in the cloud. Organizations can begin by identifying a few workloads that can be migrated to the cloud and gradually expand to larger workloads as they gain

experience. This approach minimizes risk, as organizations can learn from their experiences and adjust their approach accordingly.

Benefits: Starting small minimizes risk and allows organizations to learn from their experiences and helps build confidence and expertise in the cloud. Another benefit is that it enables organizations to realize the benefits of the cloud quickly without having to migrate all workloads simultaneously.

Use case: An organization might start migrating a few low-risk workloads, such as test and development environments, to the cloud to gain experience and confidence while minimizing the impact of any potential issues.

Adopt a phased, interactive approach

Adopting a phased approach involves dividing the migration into phases or iterations, each focusing on a specific set of workloads or applications. This approach helps manage complexity and risk, as organizations can focus on specific areas of migration and refine their approach as they go.

Benefits: This approach helps you and your organization manage complexity and risk. It also provides a structured approach to migration where the organization can learn from each phase and adjust as necessary.

Use case: An organization could divide its migration into three phases: a pilot phase, a production phase, and a final optimization phase. During the pilot phase, it could migrate a few low-risk workloads to the cloud and test their approach. During the production phase, it could migrate the bulk of its workloads to the cloud, and during the optimization phase, it could refine its technique and optimize its cloud resources.

Using a hybrid approach

A hybrid approach involves keeping some workloads on premises and migrating others to the cloud. This approach can mitigate risks and provide greater flexibility, as organizations can choose which workloads to migrate based on their specific requirements.

Benefits: This type of approach provides greater flexibility and choice. It helps mitigate risks by keeping some workloads on premises and enables organizations to leverage their existing on-premises investments while benefiting from the cloud.

Use case: An organization might keep its critical workloads on premises while migrating less critical workloads to the cloud, allowing them to mitigate risks and ensure that any issues with the cloud do not impact critical workloads.

Leverage the benefits of serverless technologies and Cloud automation

Serverless computing in Azure is a model where the cloud provider manages the infrastructure required to run the application. This means the customer pays only for the actual usage of the service. This approach can help automate modern applications by providing a scalable, event-driven, integrated architecture. With serverless computing in Azure, applications can automatically scale up or down as demand changes, trigger events to automate processes, and integrate with other Azure services.

Furthermore, leveraging automation tools and scripts can simplify the processes being performed manually and reduce the risk of error, including human error. Automation can ensure consistency, minimize manual effort, and speed up the migration process.

Benefits: Leveraging and using serverless technologies for automation on the cloud helps simplify the migration process, reduce the risk of error, and enable organizations to scale their migration efforts more efficiently.

Use case: An organization might use Azure Automation to automate the deployment of VMs and other resources in the cloud. It might also use Azure DevOps to automate the testing and deployment of applications in the cloud.

Adopt a flexible DevOps approach

Adopting a DevOps approach involves automating the deployment, testing, and monitoring of applications and infrastructure. DevOps practices can help to ensure consistency, improve efficiency, and reduce the risk of error.

Benefits: Adopting and being flexible in DevOps helps improve efficiency and reduces the risk of error, allowing for quicker and more reliable deployment and management of applications. It also creates more agility and flexibility for team collaboration.

Use case: An organization might use Azure DevOps to automate the deployment and testing of its applications in the cloud. It might also use Azure Monitor to monitor the performance and availability of its applications.

Azure Well-Architected Review

The **Azure Well-Architected Review** is a tool Azure offers to help customers assess and improve the architecture of their cloud workloads based on best practices and industry standards. It is a free, self-service tool that enables users to evaluate the readiness of their workloads in terms of security, reliability, performance efficiency, cost optimization, and operational functionality.

The review is conducted through a series of questions and assessments covering various aspects of the workload architecture. **Figure 13-5** shows how users can identify areas of improvement and receive recommendations to address issues or gaps. The tool provides users with a prioritized list of suggestions based on the impact and effort required to implement each request.

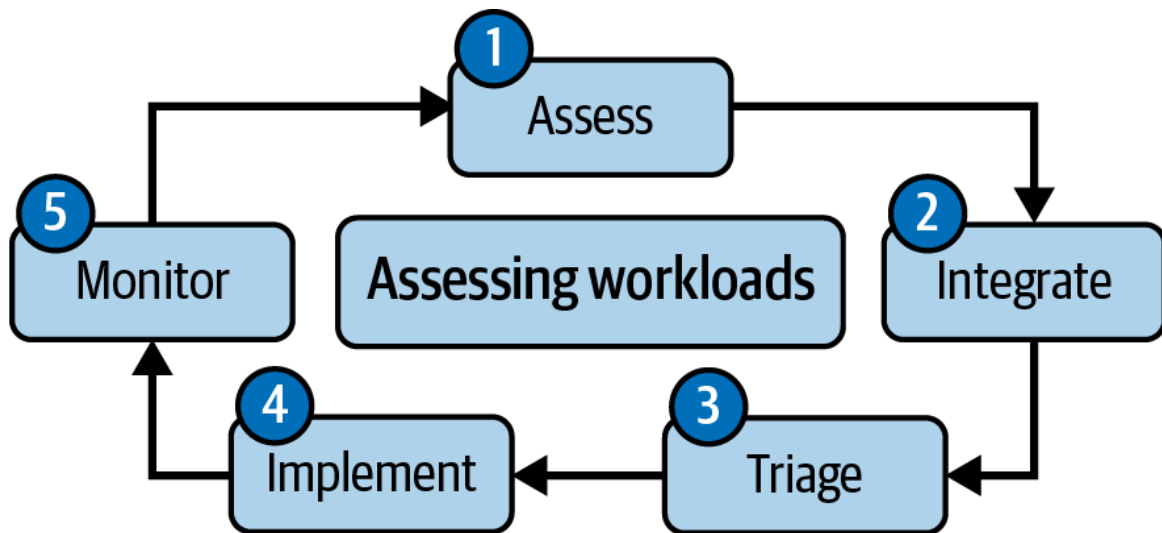


Figure 13-5. Assessing workloads with the Azure Well-Architected Review

Benefits of Azure Well-Architected Review

The Azure Well-Architected Review can provide valuable insights and recommendations to help you optimize your environment, reduce risk, and achieve your business objectives. Benefits of using a Well-Architected Review include:

Improved workload performance and efficiency

The review identifies potential bottlenecks, performance issues, and areas for optimization, resulting in enhanced workload performance and efficiency.

Reduced security risks

The review identifies security risks and provides recommendations to address them, resulting in improved security posture and reduced risk of security incidents.

Lower costs

The review identifies cost optimization opportunities and provides recommendations to reduce costs without compromising performance or security.

Improved operational excellence

The review guides improving operational processes and procedures for efficiency, resilience, and scalability.

NOTE

Many companies have used the Well-Architected Framework to improve their business and IT infrastructure on the cloud.

Coca-Cola HBC uses this framework and review to optimize its Azure environment and reduce costs. They optimized costs by implementing recommendations such as resizing VMs and shutting down unused resources. The study also identified potential security issues and improved their disaster recovery plan.

Additionally, Marston's, a British brewery and pub chain, used the review to improve its cloud infrastructure reliability and performance. The review identified potential bottlenecks and the best scaling and load-balancing practices.

The **Azure Well-Architected Review** has helped organizations across various industries optimize their Azure environments, reduce costs, improve performance, enhance security and compliance, and streamline operations. By implementing the recommendations provided by the review, these organizations achieved better efficiency, reliability, and scalability in their Azure infrastructure.

Microsoft Assessments for Evaluation and Review

Microsoft Assessments is a suite of tools and services that provide guidance and recommendations for organizations looking to adopt Microsoft cloud technologies, including Azure, Microsoft 365, and Dynamics 365. The assessments cover a range of areas, including security, compliance, migration readiness, and cost optimization.

These can help organizations identify their current strengths and weaknesses, develop a roadmap for cloud adoption, and prioritize areas for improvement. They can also help organizations optimize their cloud environment, reduce costs, and improve security and compliance.

Benefits of using Microsoft Assessments include:

Tailored recommendations

Provide personalized recommendations based on an organization's unique circumstances and requirements.

Best practices

Learn from Microsoft's expertise in cloud computing.

Time and cost savings

Identify improvement areas and prioritize cloud adoption efforts, resulting in time and cost savings.

Risk mitigation

Identify and mitigate risks associated with cloud adoption, such as security and compliance risks.

Continuous improvement

Track progress and identify areas for continuous improvement.

Useful Microsoft Assessments for Cloud Migration

Microsoft Assessments has a great list of tools and guides that help evaluate your current cloud workloads, applications, practices, processes, and tools.

Here are tools to consider when preparing for a cloud adoption or migration project:

Strategic Migration Assessment and Readiness Tool (SMART)

The SMART comprehensive assessment tool helps organizations assess their readiness to migrate their on-premises infrastructure to the cloud. It evaluates the organization's IT environment, including hardware, software, applications, data, and network connectivity. SMART provides a detailed assessment report that identifies areas needing improvement and recommends ways to optimize the migration process. It also helps organizations identify potential migration risks and estimate the cost of migrating to the cloud.

Azure Well-Architected Review

This assessment helps organizations ensure that their Azure workloads are designed according to best practices and meet industry standards for reliability, security, and performance. The review examines your workload through the different layers and perspectives of reliability of your workloads on the cloud, cost management and optimization, operational excellence, security, data protection, compliance, and performance efficiency, providing valuable insights for improving your Azure architecture.

Cloud Adoption Strategy Evaluator

This assessment evaluates your cloud adoption strategy and recommends options for building or advancing your cloud business case. It helps you determine the best approach to cloud adoption based on your organization's goals and needs.

Cloud Journey Tracker

This tracker helps you identify your cloud adoption path based on your needs and navigates to relevant content in the

Cloud Adoption Framework for Azure. It provides a clear roadmap for your organization's cloud adoption journey.

Azure Landing Zone Review

This assessment helps organizations review their Azure platform readiness so adoption can begin. It assesses your plan to create a landing zone to host workloads that you plan to build in or migrate to the cloud, ensuring a smooth and successful migration.

Cloud Adoption Security Review

This assessment helps organizations assess their security journey for cloud adoption, providing actionable recommendations and considerations for improving your security posture in the cloud to ensure that your organization's data and infrastructure are protected.

These are some of the most commonly used **assessment tools**; however, other assessment tools are available.

Microsoft Assessments provide valuable guidance and recommendations that can help organizations to optimize their cloud environment, reduce costs, and improve security and compliance.

Hybrid Cloud and Multi-Cloud Solutions in Azure

As organizations increasingly adopt cloud computing, they often use multiple cloud providers and on-premises infrastructure, leading to hybrid and multi-cloud environments. Deploying applications using hybrid and multi-cloud solutions provides several benefits to IT organizations including greater flexibility, better disaster recovery capabilities, and the ability to leverage multiple cloud providers for different workloads.

Azure offers numerous services and solutions to support hybrid and multi-cloud environments, making it easier for organizations to manage resources across different environments and cloud providers.

Azure Arc

Azure Arc is a hybrid multi-cloud management solution that allows organizations to manage resources across on-premises, multi-cloud, and edge environments from a single control plane. With Azure Arc, you gain the benefit of being able to manage servers, Kubernetes clusters, and applications running anywhere. A company can use Azure Arc to manage Kubernetes clusters running in Azure and AWS (Amazon Web Services) while managing on-premises Windows Server VMs. Organizations that use Azure Arc can run it on their servers on any cloud or on premises regardless if they are Windows, Linux, virtual, physical, domain-joined, and nondomain-joined. Azure Arc also supports and helps you extend existing Kubernetes clusters, on-premise SQL servers, private clouds, and more.

Azure Arc-Enabled Kubernetes

This enables organizations to run Kubernetes clusters across multiple cloud providers, including Azure, AWS, and Google Cloud Platform. With this service, organizations can manage their Kubernetes clusters from a single control plane, making it easier to deploy, scale, and manage applications across different environments.

Use case: An organization may have deployed a Kubernetes cluster on premises to manage its containerized applications. However, it is also running applications in Azure that it would like to manage with the same Kubernetes control plane. In this case, the organization can use Azure Arc-enabled Kubernetes to register its on-premises Kubernetes cluster with Azure, making it part of the same control plane as the Azure-based Kubernetes clusters. This enables it to manage all the Kubernetes clusters from a single control plane, simplifying application deployment and management across different environments.

Azure Stack

As a hybrid service, Azure Stack allows organizations to run Azure services on premises. With Azure Stack, organizations can build and run hybrid applications across Azure and on-premises environments using the same Azure management tools, APIs, and experiences.

Use case: A company can use Azure Stack to deploy a hybrid application that uses both Azure and on-premises resources. For example, the application may use Azure's cognitive services for image recognition while the backend data processing is done on premises.

Azure VMware Solution

This is a first-party Azure service that enables organizations to run VMware workloads natively on Azure. With this solution, organizations can migrate VMware workloads to Azure without refactoring or rearchitecting their applications.

Use case: Organizations that have Azure Commercial or Government licenses can make use of Azure VMware Solution. This service can be used to create and configure private clouds with clusters of VMware vSphere within the cloud infrastructure of Azure. Azure VMware follows the Shared Responsibility model between Microsoft and its users. To learn more about the specific matrix of this responsibility model, please read [“Azure VMware Solution responsibility matrix: Microsoft vs. customer”](#).

Azure ExpressRoute

One of the networking services discussed in [Chapter 4](#), ExpressRoute is a service that provides dedicated, private network connectivity between Azure data centers and on-premises infrastructure. This service enables organizations to extend their on-premises network into Azure, providing a hybrid cloud solution.

Use case: Since this service uses a private connection and industry-standard secured dynamic routing protocols, Azure ExpressRoute can be used by

organizations who want to transmit their enterprise data on a private and secured network. Another example use case is migrating on-premises systems to Microsoft Azure on a large scale.

Azure Site Recovery

Azure Site Recovery enables organizations to replicate and failover VMs and applications between on-premises data centers and Azure, providing disaster recovery capabilities for hybrid environments.

Use Case: A company may have an application running on premises critical to its business operations. However, if there is a disaster, such as a fire or a flood, the on-premises infrastructure may become unavailable, causing significant downtime for the application. In this case, the company can use Site Recovery to replicate the application to Azure, providing continuity of service while it restores its on-premises infrastructure. This provides a cost-effective and reliable disaster recovery solution that minimizes downtime and ensures business continuity.

Azure VPN Gateway

This solution enables organizations to establish secure, encrypted connections between on-premises networks and Azure. With this service, organizations can extend their on-premises network into Azure, enabling them to access Azure resources over the internet securely.

Use case: An organization may have a hybrid environment with resources deployed on premises and in Azure. To access Azure resources securely, it can use Azure VPN Gateway to establish a secure, encrypted connection between its on-premises network and Azure. This enables the organization to extend its on-premises network into Azure, making it easier to manage resources and access them securely. This is particularly useful for organizations that need to access resources that are not exposed to the public internet or that have strict security requirements.

Multi-Cloud and Hybrid Solutions in Azure

Azure also offer solutions that enable organizations to run workloads across multiple cloud providers, including Azure, AWS, and Google Cloud Platform. These solutions include Azure Arc-enabled Kubernetes. It enables organizations to run Kubernetes clusters across multiple cloud providers, and Azure API Management, which enables organizations to manage APIs across multiple cloud providers.

Use case: Hybrid and multi-cloud solutions enable remote work within organizations by providing a seamless, flexible, and secure approach to access data and applications globally. It also allow organizations to take advantage of cloud innovation. They can also take advantage of the backup and disaster recovery options for IT resources across on-premises, multi-cloud, and edge environments. Azure Arc helps extend on-premises, edge, and cloud applications in different cloud environments.

In terms of infrastructure as code (IaC) Azure Terraform, organizations can build their infrastructure and cloud resources and deploy to different cloud providers using automation and infrastructure as code with Terraform for Azure.

A company may use Azure Arc-enabled Kubernetes to run a Kubernetes cluster across both Azure and AWS, enabling them to leverage the benefits of both cloud providers for different workloads.

These are just a few examples of how Azure Arc-enabled Kubernetes, Azure Site Recovery, and Azure VPN Gateway can be used in different scenarios. There are many other ways that these services can be used to support hybrid and multi-cloud environments, depending on the organization's specific needs.

Overall, hybrid and multi-cloud solutions for Azure offer several benefits to organizations, including greater flexibility, better disaster recovery capabilities, and the ability to leverage multiple cloud providers for different workloads.

Learn By Doing (Try It!)

The following are recommended tutorials from Microsoft's official documentation:

- [Strategic Migration Assessment and Readiness Tool \(SMART\)](#)
- [How to accelerate migration using CAF](#)
- [Building your DevOps practice based on the Cloud Adoption Framework for Azure](#)
- [Monitor your hybrid and multicloud machines through Azure Arc-enabled servers](#)
- [Azure Stack HCI foundations](#)

Summary

In this chapter, we learned more about cloud application modernization and adoption. We learned that cloud adoption and modernization in Azure involves moving existing applications and infrastructure to the cloud and leveraging cloud-native services to optimize and modernize applications.

Azure offers a range of services and solutions to support cloud adoption and modernization, including the Cloud Adoption Framework (CAF), the Well-Architected Framework (WAF) for Azure, and a range of hybrid and multi-cloud solutions.

The Cloud Adoption Framework is a comprehensive guide to help organizations adopt and modernize their applications in Azure. It guides planning, building, and managing cloud solutions using Azure best practices. It also includes some good-to-know anti-patterns to help organizations avoid common mistakes when adopting and modernizing applications in Azure.

In addition to the Well-Architected Framework and Microsoft Assessments tools, we also learned about the core components of the CAF for Azure,

including strategy, plan, readiness, adoption, governance, management, and security. These components provide a structured approach to cloud adoption and modernization, helping organizations to plan, implement, and manage cloud solutions effectively.

Check Your Knowledge

1. What is the Cloud Adoption Framework for Azure, and how can it help organizations with cloud adoption and modernization?
2. How can Microsoft Assessments help organizations with cloud adoption and modernization?
3. How can the Well-Architected Framework for Azure help organizations optimize their cloud solutions?
4. What hybrid and multi-cloud solutions are available in Azure, and how can they help organizations?
5. How can Azure Arc-enabled Kubernetes help organizations with hybrid and multi-cloud environments?

Answers to these questions are in the [Appendix](#).

Recommended Learning Resources

“Azure Application Architecture Fundamentals.” Microsoft Learn, December 16, 2022, <https://oreil.ly/BT6uZ>.

“Azure Arc Documentation.” Microsoft Learn, <https://oreil.ly/n1IMU>.

“Azure Architecture Center.” Microsoft Learn, <https://oreil.ly/pV8Az>.

“Azure Cloud Migration Best Practices Checklist.” Microsoft Learn, July 7, 2023, <https://oreil.ly/WLRnj>.

“Azure Migration Guide Overview.” Microsoft Learn, July 7, 2023, <https://oreil.ly/2Ys9S>.

“Cloud Design Patterns.” Microsoft Learn, April 13, 2023,
<https://oreil.ly/kPXgH>.

“Designing Reliable Azure Applications.” Microsoft Learn, August 8, 2023,
<https://oreil.ly/DJ102>.

“Hybrid Architecture Design.” Microsoft Learn, December 16, 2022,
https://oreil.ly/_yy8q.

“Microsoft Azure Well-Architected Framework.” Microsoft Learn, March 27, 2023, *<https://oreil.ly/wB-DU>*.

“Microsoft Cloud Adoption Framework for Azure.” Microsoft Learn,
<https://oreil.ly/aAk8z>.

Chapter 14. Cloud Development Tools for Azure

Every software success story starts with developers shaping their dream and idea into an executable. This application first becomes available in a local development environment before it grows and matures for broader access through the prism of software development platforms. Having meaningful, purpose-built cloud development tools is essential for developers' productivity, efficiency, and time-saving. The availability of these tools empowers developers to build genuinely cloud-native applications that are well-optimized, resource-efficient and sustainable—with ease. Getting an overview of cloud development tools for Azure and ways you can use them to their max potential in your project will make you a true Azure Developer Hero!

—Kristina Devochko, Principal Cloud Engineer at Amesto
Fortytwo, Microsoft Azure MVP, CNCF Ambassador,
Kubernetes Unpacked podcast host, and content creator at
kristhecodingunicorn.com

Now that you have learned about cloud services in Azure across all categories, you should be ready to develop applications and solutions and solve problems using your favorite programming languages.

In this final chapter of the book, you will learn about the different cloud development tools, DevOps tools, and recommended Microsoft Certifications. By understanding these and adding them to your toolbox as a developer and DevOps engineer, you are prepared to design, architect, and develop robust and reliable cloud applications whether they are fully cloud-native, hybrid, or multi-cloud solution projects.

By the end of this chapter, you will be prepared to design, architect, and develop with Azure on any cloud migration or modernization projects in your organization.

Importance of Development Tools for Developer Productivity

A **study** conducted by JetBrains in 2019 found that developers who use integrated development environments (IDEs) are more productive than those who use text editors. In this survey, developers stated that using IDEs has helped them complete tasks faster and with fewer errors. IDEs also help developers improve code quality using code refactoring tools.

TIP

Microsoft published a research **article** on developer productivity. They called the concept the **SPACE of Developer Productivity**, which is based on productivity analysis. These articles demystify some myths; they also discuss why proper developer tools are relevant to understanding developer productivity and getting more metrics to guide their work and teams.

Effective development, programming, and team collaboration tools are essential for developers to maximize their productivity and focus, for these reasons:

Efficiency

Tools help developers work more efficiently. For example, a text editor with features like code highlighting, autocomplete, and search functions helps developers write code faster with fewer errors. Similarly, a debugger can help them quickly find and fix bugs in their code.

Focus

Tools can also help developers stay focused on their work. For example, a distraction-free text editor can minimize interruptions and allow developers to focus on their code. Similarly, a task management tool can help developers

prioritize their work and avoid getting sidetracked by other tasks.

Collaboration

Tools for pair programming and live coding with colleagues can help developers deliver their tasks. Developer collaboration and communication tools can also make working with other teams easier. For example, version control tools like Git, Azure DevOps, GitHub, and live sharing features in IDEs allow multiple developers to work on the same codebase without causing conflicts. Similarly, communication tools like Slack can help developers stay in touch more efficiently than with email and collaborate effectively.

Consistency and coding standards

Coding standards within teams can help ensure consistency in coding practices and styles, which can be important for large projects or teams. For example, a code formatting tool can help ensure all code is formatted consistently across the project, making it easier to read and maintain.

Although developer productivity is hard to measure and is individual, effective development and programming tools are crucial for developers. Developer tools help them work efficiently, stay focused, collaborate effectively, and maintain consistency. It can help them save time, avoid errors, and produce higher-quality code.

In the rest of this chapter, you will learn the different tools for cloud development with a focus on Azure.

Azure Development Tools for Engineers

The developer tool you use is typically determined by your role in your IT project, the systems or applications you are working with, and the cloud provider for those systems.

Development and DevOps tools are crucial for developers and engineers who work with Azure because they offer a comprehensive IDE with advanced features, such as debugging, testing, and deployment automation. These tools foster seamless collaboration among team members regardless of location and provide a secure and scalable infrastructure for building and deploying applications.

Moreover, development and DevOps tools for Azure are designed to integrate with other platforms, APIs, SDK, and other devices, enabling developers and engineers to leverage the full capabilities of these resources to build and deploy applications efficiently, securely, and quickly.

Visual Studio and Visual Studio Code

Visual Studio is an all-encompassing IDE that gives programmers powerful tools for creating programs for various platforms and technologies, including Azure. With this comprehensive IDE, developers can write, debug, and deploy code in any operating system or environment. **Figure 14-1** illustrates the versions available within the **Visual Studio family**.

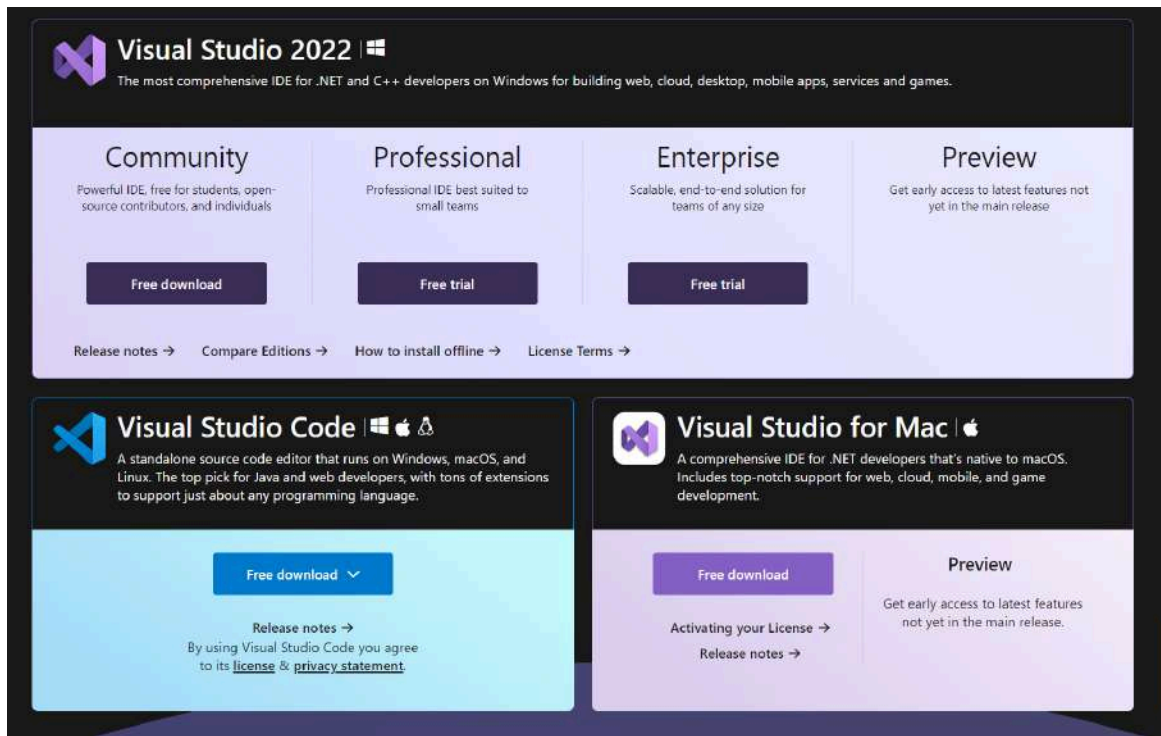


Figure 14-1. The Visual Studio family

This IDE enables developers to construct applications for Azure with less time and cost. Because the IDE includes support for Azure development, developers can easily create, manage, and deploy Azure resources directly from within the Visual Studio environment. As a result, it is much simpler for developers to concentrate on creating code and constructing applications rather than fretting over the underlying infrastructure and configurations.

Following are the benefits of using Visual Studio for the creation of Azure applications:

Integrated support for a wide range of programming languages

Support for programming languages includes C#, JavaScript, Python, and others. Because of this, developers don't need to learn a new programming language or tool to create applications for Azure; they can use their favorite language.

Robust debugging and testing capabilities

Visual Studio has robust debugging and testing capabilities that enable developers to find errors in their code and solve them quickly. Additionally, the IDE offers highly developed profiling tools that allow developers to enhance the performance of Azure applications.

Preconfigured templates and tools

The IDE has various preconfigured templates and wizards, making creating Azure resources such as VMs, storage accounts, and databases easier. This makes it simpler for developers to begin working on Azure projects and reduces the time and effort needed to create complicated apps.

In addition, Microsoft recently released a new tool for Visual Studio called **IntelliCode**, an AI-powered coding helper that assists developers as they create code to make it more efficient and effective. The numerous capabilities help with tasks such as code completion, code recommendations, and code snippets and are driven by ML algorithms that examine millions of lines of code.

Integration with the AI technology provided by **GitHub Copilot** is one of the most compelling aspects of IntelliCode. Copilot uses ML techniques to analyze millions of lines of code from open source repositories hosted on GitHub to deliver real-time intelligent code recommendations and completions.

Developers may use GitHub Copilot as a coding or pair-programmer assistant while writing code in Visual Studio or VS Code. It suggests finishing code snippets, functions, and classes according to the code context the developer writes. This tool uses natural language processing to attempt to comprehend the developer's purpose and offer suggestions tailored to the developer's preferred coding style and aesthetic.

Junior developers who are fresh to any programming language or framework or working on a project with a complex codebase may find

Copilot a beneficial tool. It can assist developers in writing code more quickly and precisely, reducing the time and effort required to perform coding jobs. It is possible to integrate Copilot with IntelliCode in Visual Studio.

TIP

You must meet the minimum computer requirements to install IntelliCode in Visual Studio. If you are using VS Code, there is an extension as well.

Visual Studio has various editions and types, and you can select one appropriate for your needs and goals:

Visual Studio Community

This is a free, fully featured edition of Visual Studio designed for individual developers, open source projects, academic research, and learning in the classroom.

Visual Studio Professional

This paid edition of Visual Studio is meant for professional developers who want additional features and tools for producing high-quality programs for platforms including Windows, Android, iOS, and the web. It supports complex frameworks and technologies and advanced debugging and testing tools. Additionally, it has features that facilitate cooperation.

Visual Studio Enterprise

This is the complete edition of Visual Studio created with enterprise development teams in mind. It comes with all of the features in Visual Studio Professional and additional tools for testing, profiling, and optimizing applications. In

addition, it provides highly developed collaboration and DevOps capabilities for groups working on large-scale projects.

Visual Studio Code

VS Code is a simple, cross-platform code editor intended for programmers who favor a more streamlined experience when they are programming. It is compatible with various programming languages and frameworks and has built-in support for Git, extension development, and debugging. Using the different extensions for VS Code, you can personalize and customize your development tools for local development to assist with your day-to-day productivity. The **C# Dev Kit VS Code extension** is recommended if you are using VS Code and coding with C# (currently in Preview version at the time of writing).

In addition to these editions, Visual Studio also comes in specialized forms supporting different types of programming or development, including creating video games, AI research, ML, data science, and other exciting solutions with the cloud technologies on Microsoft Azure. **Visual Studio for Unity**, used for game development, VS Code extensions for data science development, and Visual Studio Tools for AI are all examples of these specialized versions of this IDE that help developers work with different types of projects. The many versions of Visual Studio give developers many options, which are aligned with developer or team requirements.

In a nutshell, the Visual Studio family is incredibly helpful software for any developer who works with Azure. It provides a complete collection of tools and features that make creating, managing, and deploying apps built on Azure simpler. This robust, established IDE can assist developers in building Azure apps more effectively and efficiently while simultaneously decreasing the time and effort needed to create complicated applications.

Alternative IDEs for Java, Cross-Platform, or Mobile Development

Aside from Microsoft's integrated IDEs for Azure, developers can use other IDEs and tools to develop and deploy applications on the Azure platform. The following examples focus on Java and mobile development.

JetBrains Rider

Many .NET developers building cross-platform applications also use **JetBrains Rider** as their IDE for programming. It is actually a .NET-based IDE on IntelliJ with ReSharper.

Eclipse

Eclipse is a popular open source IDE that supports various programming languages, including Java, C/C++, and Python. To use Eclipse for Azure development, you can install the **Azure Toolkit for Eclipse**, which integrates with Azure services. The toolkit includes wizards and templates that make it easy to create, deploy, and manage Azure resources directly from Eclipse.

IntelliJ IDEA

IntelliJ IDEA is a popular IDE for Java development. To use IntelliJ IDEA for Azure development, you can install the Azure Toolkit for IntelliJ, which integrates with Azure services such as Azure Functions and Spring Cloud Azure. The **Azure Toolkit for IntelliJ** includes wizards and templates that make it easy to create, deploy, and manage Azure resources directly from IntelliJ IDEA.

Android Studio

Android Studio is the official IDE for Android app development. To use Android Studio for Azure development, use the **Azure SDK for Android**, which integrates with Azure services.

Xcode

Xcode is the official IDE for iOS app development. To use Xcode for Azure development, install the **Azure SDK for iOS**. It integrates with Azure and contains libraries and tools that make it easy to create, deploy, and manage Azure resources directly with Xcode.

Azure Software Development Kits

Azure Software Development Kits (Azure SDKs) are collections of libraries, tools, and documentation that enable developers to build applications that interact with Azure services. The SDKs provide a convenient way for developers to access the features and functionality of Azure services from within their applications.

Azure SDKs support a wide range of programming languages, including .NET, Java, Node.js, Python, and Ruby. They also provide a consistent interface for developers to work with Azure services, regardless of the language they are using.

Critical uses of Azure SDKs for cloud development include:

Simplifying development

Azure SDKs provide prebuilt libraries and tools for developers to build Azure applications quickly and easily. SDKs offer additional functionality that developers can leverage to create more robust and feature-rich applications.

Ensuring compatibility with different platforms

Azure SDKs are designed to work seamlessly with Azure services, ensuring that applications built with them are compatible and can take full advantage of the latest features.

Improving performance

SDKs often include optimizations and best practices that can help developers build more efficient and performant applications.

Streamlining deployment

Many Azure SDKs include deployment tools that simplify deploying applications to Azure and managing their lifecycle.

Azure SDKs support a wide range of programming languages. They allow developers to build applications using C#, VB.NET, and F#. The Azure SDK for .NET includes libraries and additional support for .NET Framework and .NET Core applications. The Java SDK enables developers to build applications using Java and supports various Azure services. If you're developing JavaScript-based applications with Azure, the Node.js SDK provides libraries. Python, Ruby, Go, and PHP developers can interact with Azure services from any programming language that supports HTTP requests. **Azure REST APIs** can interact with Azure services from any programming language that supports HTTP requests.

TIP

The **Azure SDK Releases** homepage lists all SDK, code libraries, and documentation for languages such as .NET, Java, Go, JavaScript/Typescript, and others.

Azure SDKs are a valuable resource for developers looking to build applications that leverage the power and flexibility of Azure services. They simplify development, enhance functionality, ensure compatibility, improve performance, and streamline deployment, making it easier for developers to create robust and scalable applications.

Azure Command-Line Tools

In Azure, automation is standard. Any time you develop, manage resources, and interact with Azure's platform, it executes code to retrieve, update, remove data, and perform other cloud operations.

You can choose from a variety of resource management approaches for Azure. The following sections discuss the different command-line tools for Azure development. These tools allow developers to use Azure's preexisting automation.

Azure Cloud Shell

Azure Cloud Shell is a web-based shell environment that allows you to manage your Azure resources from anywhere. It provides access to the Azure command-line interface (CLI) (with Bash) and Azure PowerShell command-line tools without needing to install them locally. You can use Azure Cloud Shell to manage your Azure resources, create and test scripts, and perform other administrative tasks. For example, in Azure Cloud Shell, you can create an Azure VM or any resource within minutes.

To create an Azure resource using the command-line tool, you must perform these steps.

1. Connect to Azure using Azure CLI or Azure PowerShell on Azure Portal or in a local development environment, as shown in **Figure 14-2**.
2. Authenticate to Azure by logging in with the `az login` command. You don't need to do this if you are already logged in on Azure. Otherwise, you need to run this command with your username and password. Aside from standard Microsoft Entra ID users, you can log in using your service principal and managing identities.
3. Choosing the right Azure subscription to work with is essential. If you are still determining what Azure subscription is set as default, you can run the command `az account show` to choose the resulting output to different formats such as JSON, Table, YAML, or TSV.

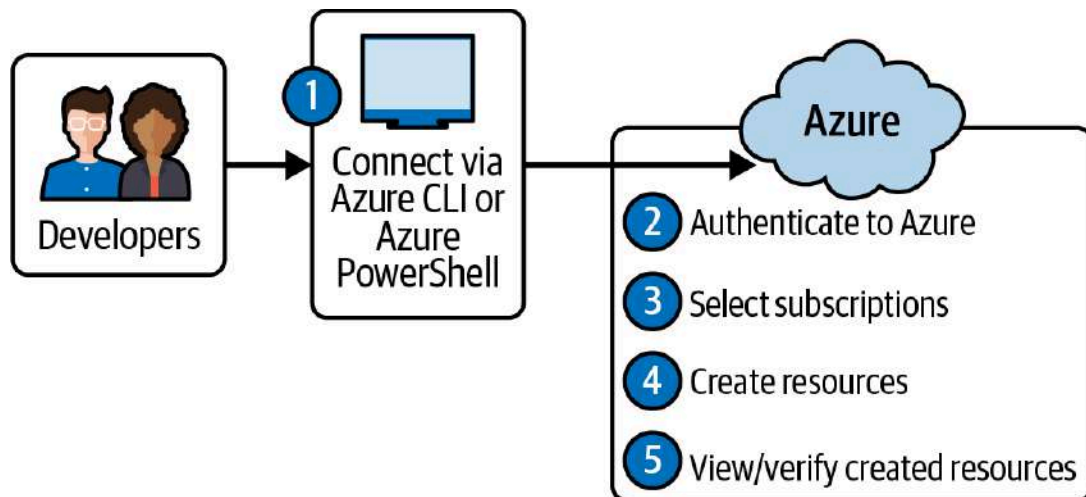


Figure 14-2. Developers connect and manage Azure resources using Azure Cloud Shell (Azure CLI or Azure PowerShell)

4. Start creating or managing resources. First, you typically create a resource group to create your Azure resources using the following command:

```
az group create --name <name of resource group>
--location <azure region or location>
```

You can verify if creating a new resource group was successful because it will tell you on the command line. Alternatively, you can also query using the command `az group list` or further specify by setting the specific name of the resource group you select.

For example, you can use this command to specifically query the name of the resource group:

```
az group list --query "[?name == '<resource group name>']"
```

Launching Azure Cloud Shell creates a temporary storage account to store your session data, scripts, and files. This storage account is used to persist your data across Cloud Shell sessions and to ensure that your data is accessible from any machine or browser you use to access Azure Cloud Shell.

The temporary storage account uses Azure Files, which provides fully managed file shares in the cloud. This means your data is stored securely in Azure and can be accessed from anywhere with an internet connection. Your files are also encrypted at rest and in transit, ensuring the security of your data.

Azure Cloud Shell also supports built-in code editors that allow you to create, modify, and save files directly from the command line, even on Azure Portal, through a web browser. The editors that are supported include Vim, Nano, and Emacs. You can use other editors like Visual Studio Code, Sublime Text, or Atom by installing them within your Cloud Shell session.

Azure Command-Line Interface (CLI)

Azure CLI is a cross-platform command-line tool that enables developers to manage and interact with Azure resources. It provides a CLI for Azure services and allows developers to perform various tasks, such as creating and managing VMs, storage accounts, databases, etc. It is built on top of the Azure Resource Manager API, allowing it to support all its services. Developers can use Azure CLI to perform tasks typically done through the Azure Portal, PowerShell, or other Azure tools. Azure CLI supports different Azure services for AI, ML, App Service, Azure Functions, Azure VM Scale Sets, Service Fabric, and more.

Benefits of using Azure CLI for cloud development and deployments include:

Platform independence

Since this is a cross-platform tool, you can use it flexibly, and it can be run on Windows, Linux, and macOS.

Automation

Developers can use Azure CLI to automate various tasks and create scripts for the deployment and management of Azure resources.

Faster development cycles

With Azure CLI, developers can perform tasks quickly and efficiently, speeding up the development cycle.

Flexibility

Azure CLI allows developers to work with Azure resources comfortably. It supports various command-line shells, including Bash, PowerShell, and Windows Command Prompt.

Integration with other tools

Azure CLI integrates well with other devices, such as Terraform, Ansible, and Jenkins, making it easier to integrate Azure into the development workflow.

Azure CLI is a valuable tool for developers working with Azure. It offers many advantages over the Azure Portal in specific scenarios, such as scripting and automation, remote access, rapid development, advanced configuration options, and integration with another tool.

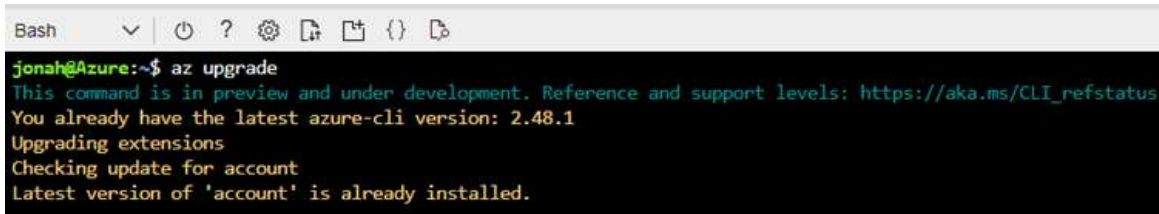
TIP

Suppose you are new to learning Azure; learning Azure CLI commands can be helpful when you want to create and manage your Azure resources. The Azure CLI can be accessed using the Azure Cloud Shell on the Azure Portal by typing *shell.azure.com* on your web browser. If you manage resources on the go using the Azure mobile app, you can select the menu option for Azure Cloud Shell.

Another option is to **install** it on your computer locally. It supports Windows, Linux, and macOS; you can also run these Azure CLI commands on **Docker**. Learn about the common **CLI commands** in the official Azure CLI guide.

It is essential to note that the Azure CLI allows you to configure how you prefer your upgrades. You can manually update using the CLI command **az**

upgrade, as shown in [Figure 14-3](#). You can also configure it to automatically upgrade using the CLI command `az config set auto-upgrade.enable=yes`.



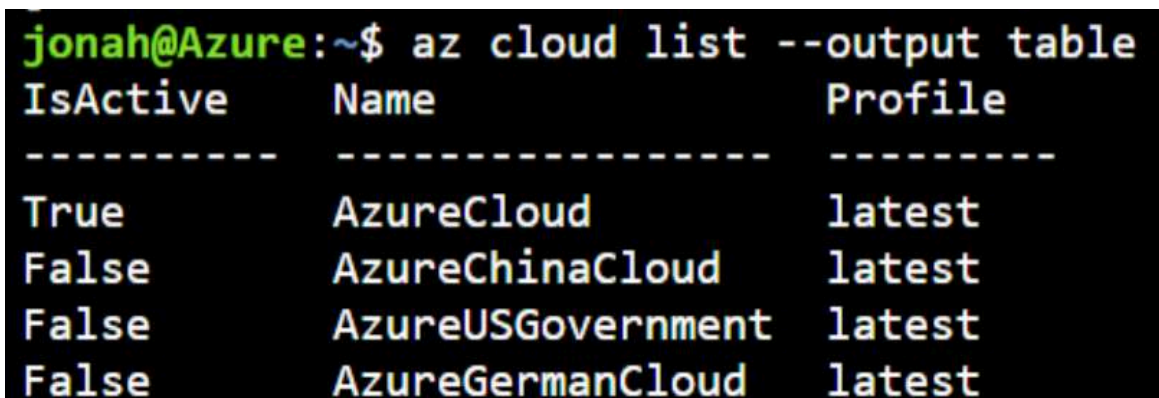
```
Bash
jonah@Azure:~$ az upgrade
This command is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
You already have the latest azure-cli version: 2.48.1
Upgrading extensions
Checking update for account
Latest version of 'account' is already installed.
```

Figure 14-3. Manual upgrade of Azure CLI using command `az upgrade`

In addition to managing resources using Azure CLI, you can work with multiple cloud environments through these command-line tools.

The command `az cloud list` helps you identify which cloud is active or enabled, as shown in [Figure 14-4](#). When you need to switch cloud environments, you can use the following command:

```
az cloud set --name <the name of the cloud> you want to change to>
```



```
jonah@Azure:~$ az cloud list --output table
IsActive      Name                Profile
-----
True          AzureCloud          latest
False         AzureChinaCloud     latest
False         AzureUSGovernment   latest
False         AzureGermanCloud    latest
```

Figure 14-4. List of the active cloud you have enabled

Note that you can have only one cloud active at the same time. Learn more about managing other types of clouds within Azure using the Azure CLI. Check out how you can [manage registered clouds](#) using Azure CLI commands.

WARNING

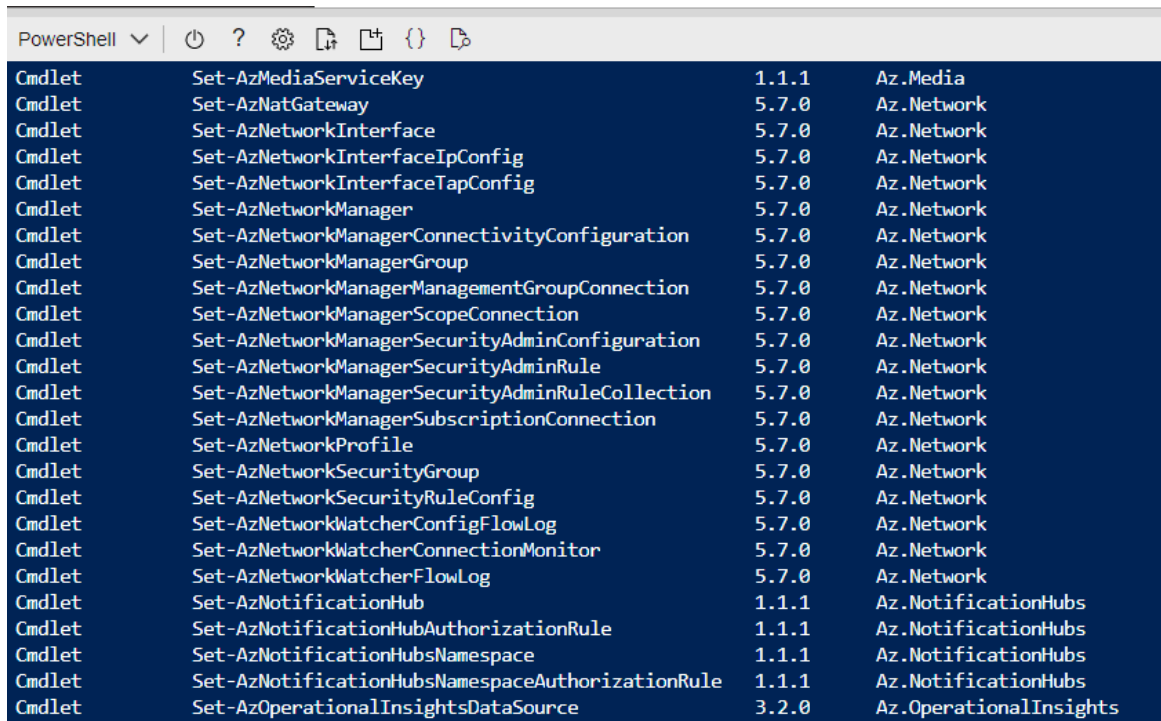
Before deploying the Azure CLI, you must add specific IP addresses and domain URLs to the allow list if a network firewall or proxy server protects your organization. Check out Microsoft's list of endpoints for Azure CLI that you can use for different types of registered clouds in configuring your proxy servers.

There is much to learn about Azure CLI and its commands. Azure CLI is a powerful tool that can help developers manage and interact with Azure resources efficiently, automate tasks, and improve their development workflow.

Azure PowerShell

Azure PowerShell is a command-line tool that provides cmdlets (native PowerShell commands) for managing Azure resources and services (Figure 14-5). It allows you to automate everyday Azure tasks using PowerShell scripts. Azure PowerShell is great for managing complex Azure environments with multiple resources.

Azure PowerShell and Azure CLI are both command-line utilities but there are significant distinctions between the two cloud terminal tools.



Cmdlet	Set-AzMediaServiceKey	1.1.1	Az.Media
Cmdlet	Set-AzNatGateway	5.7.0	Az.Network
Cmdlet	Set-AzNetworkInterface	5.7.0	Az.Network
Cmdlet	Set-AzNetworkInterfaceIpConfig	5.7.0	Az.Network
Cmdlet	Set-AzNetworkInterfaceTapConfig	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManager	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerConnectivityConfiguration	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerGroup	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerManagementGroupConnection	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerScopeConnection	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerSecurityAdminConfiguration	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerSecurityAdminRule	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerSecurityAdminRuleCollection	5.7.0	Az.Network
Cmdlet	Set-AzNetworkManagerSubscriptionConnection	5.7.0	Az.Network
Cmdlet	Set-AzNetworkProfile	5.7.0	Az.Network
Cmdlet	Set-AzNetworkSecurityGroup	5.7.0	Az.Network
Cmdlet	Set-AzNetworkSecurityRuleConfig	5.7.0	Az.Network
Cmdlet	Set-AzNetworkWatcherConfigFlowLog	5.7.0	Az.Network
Cmdlet	Set-AzNetworkWatcherConnectionMonitor	5.7.0	Az.Network
Cmdlet	Set-AzNetworkWatcherFlowLog	5.7.0	Az.Network
Cmdlet	Set-AzNotificationHub	1.1.1	Az.NotificationHubs
Cmdlet	Set-AzNotificationHubAuthorizationRule	1.1.1	Az.NotificationHubs
Cmdlet	Set-AzNotificationHubsNamespace	1.1.1	Az.NotificationHubs
Cmdlet	Set-AzNotificationHubsNamespaceAuthorizationRule	1.1.1	Az.NotificationHubs
Cmdlet	Set-AzOperationalInsightsDataSource	3.2.0	Az.OperationalInsights

Figure 14-5. Azure PowerShell cmdlets

One noticeable difference is that Azure PowerShell utilizes the PowerShell scripting language, whereas Azure CLI uses multi-platform tools based on Node.js. This means that developers familiar with PowerShell will find Azure PowerShell more familiar and straightforward. In contrast, those who are familiar with a command-line interface more reminiscent of Linux may prefer Azure CLI.

In addition, the syntax and structure of the commands differ. Azure PowerShell commands are typically longer and more verbose than other PowerShell commands, using a similar noun-verb syntax. In contrast, Azure CLI commands are shorter, utilizing a verb-noun syntax identical to other Linux-style commands.

As noted, PowerShell commands are called cmdlets and they allow you to execute any command installed on your computer. Cmdlets are not independent executable files; instead, they are native PowerShell commands. PowerShell modules are collections of cmdlet scripts that can be invoked whenever required. To view the list of different cmdlets

available for Azure PowerShell, you can run the command `Get-Command -Module Az`.

As noted earlier, the names of PowerShell's cmdlets are always a combination of a verb and a noun. For example, the `Get-Command` cmdlet can be used to attempt to retrieve all the cmdlets registered in the command shell. The noun indicates the resource the cmdlet uses to carry out the activity that the verb describes, and the verb describes the operation that the cmdlet carries out using that resource.

Whether you choose Azure PowerShell or Azure CLI will depend on individual tastes and expertise. Both tools provide potent and versatile methods for handling Azure resources from the terminal.

NOTE

Azure CLI and Azure PowerShell are powerful tools for managing Azure resources, but they have differences in platform support, command syntax, scripting and automation capabilities, user experience, and functionality. Developers and administrators should choose the best tool for their needs and expertise.

Predictive IntelliSense in Azure Cloud Shell

The Predictive IntelliSense feature of Azure Cloud Shell is enabled by **PSReadLine**, which is used in Azure PowerShell. If you install and activate the Azure PowerShell predictor module **Az.Tools.Predictor**, it will look like Figure 14-6.

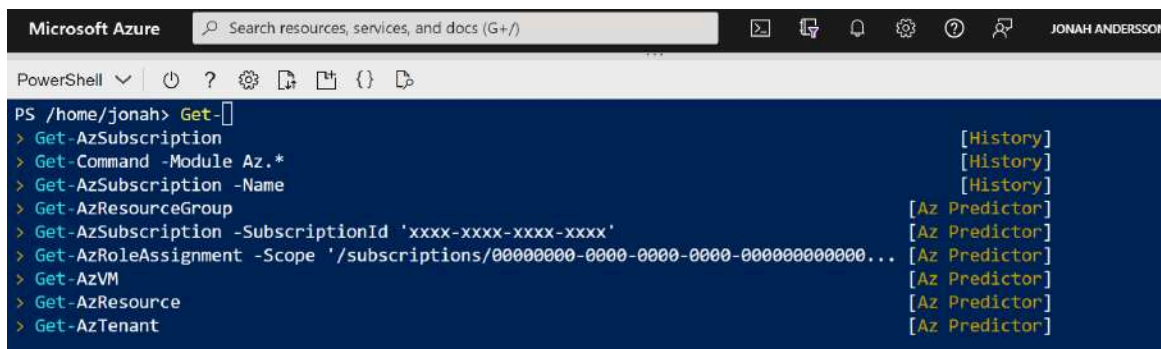


Figure 14-6. Azure Cloud Shell's Predictive IntelliSense

An assistant that suggests commands is an excellent way to increase productivity. It improves the experience of using the command line by making suggestions available that assist new as well as experienced Azure users in discovering, editing, and executing complete commands.

You should follow best practices for using Azure CLI and PowerShell, including using variables to simplify scripts and commands, using comments to document your code, testing scripts and orders in a nonproduction environment before deploying to production, using version control to manage your code and scripts, and implementing safety and compliance practices. It's also ideal to stay current with the latest [Azure CLI and PowerShell releases](#).

Azure Developer CLI (azd)

[Azure Developer CLI](#) (azd) is a valuable tool for developers that helps them accelerate their development work and deployment of applications to Azure. The features of azd deliver developer-friendly commands for critical workflow stages in the terminal, editor, IDE, or CI/CD.

azd's extensible templates contain everything required to run an Azure application. Reusable infrastructure and proof-of-concept code for applications are included in these templates. You can design a template or create one. To implement and use this tool in your development projects, you must install the latest version of recommended azd tools on your local machine, depending on your platform.

In addition, there is an alternative way to use azd on your development containers, such as DevContainers and VS Code, using the Remote Containers extension and GitHub Codespaces. Any of these have advantages and disadvantages, so check their limitations and choose the appropriate ones based on your use case.

Standard programming languages, such as .NET, Java, Python, and NodeJS (as of the date of writing), are supported for the azd. Note that this service's

features were released in phases; please review the guidance on strategies for feature versioning and release.

There are many Azure services that you can use with the service's features. For example, you can host your applications using **feature flags**. You can use supported Azure services for computing, such as the Azure App Service, Azure Container Apps, Azure Functions, Azure Static Web Apps, and other services.

Following is an example of the azd workflow for a developer (the azd tool has been installed locally on dev containers or GitHub Codespaces):

1. Decide and choose the boiler or starter template for Azure Developer CLI (azd).
2. Run the command `azd init` to initialize the template project. You typically run this as you set up your environment. If you have created one, you view or manage your environments on this tool by running `azd env`.
3. Run command `azd up` to build, package, deploy and provision the application.
4. Code changes on your applications.
5. Deploy changes to your applications using the command `azd deploy`.

If you need to update the infrastructure code on Bicep or Terraform, you can use the command `azd provision*`. You can use the command `azd monitor` for viewing the monitoring and metrics. The command `azd up - debug` can also be used for troubleshooting and debugging.

The **azd templates** are blueprint code repositories that help developers create templates. They include different folders to get started and can be customized depending on the use case. For example, in the structure of these azd templates, you will find the application code available in multiple programming languages, IaC, which can be Bicep or Terraform code, and other essential components. These templates can be customized for your

application and used with the Azure Developer CLI (azd) to deploy them to Azure. The *azure.yaml* schema describes the apps and Azure resources in these *azd* templates.

Microsoft Dev Box

Microsoft Dev Box is a cloud-based developer workstation designed for developers who interact with Microsoft technologies and intend to develop, test, and deploy applications on Azure.

Using Microsoft Dev Box for cloud development on Azure offers these benefits:

High performance

Dev Box offers an efficient machine with robust components such as a fast processor, a lot of memory, and high-speed storage, enabling developers to run multiple VMs, containers, and services on the same machine and test applications quickly and efficiently.

Built-in development tools

This cloud-based environment for development teams includes preconfigured Azure tools, such as Visual Studio, Azure CLI, Azure PowerShell, and other development tools to facilitate the development experience and accelerate application development and deployment on Azure.

Accelerated development

By using a dev box, developers can reduce the time required to build a development environment, as the dev box is preconfigured with all the tools, libraries, and frameworks. This enables developers to begin developing and testing their applications immediately, which can lead to quicker application delivery and increased productivity.

Additionally, by using Microsoft Dev Box, developers can easily collaborate by sharing their development environment, which enables developers and team members who are part of the development process to work on the same codebase without conflicts.

It can also be a cost-effective solution for developers who want to start developing and testing their applications on Azure without investing in costly hardware or infrastructure. Before a new developer can start the work, they must set up their local development environment. Sometimes they must wait for their development laptops, which can take time, especially working remotely from home.

With a dev box, developers can use a single workstation for development, testing, and deployment, which can reduce the overall cost of the product.

Microsoft Dev Box includes several components, including the Dev Center, which serves as a logical container on Azure, as shown in **Figure 14-2**.

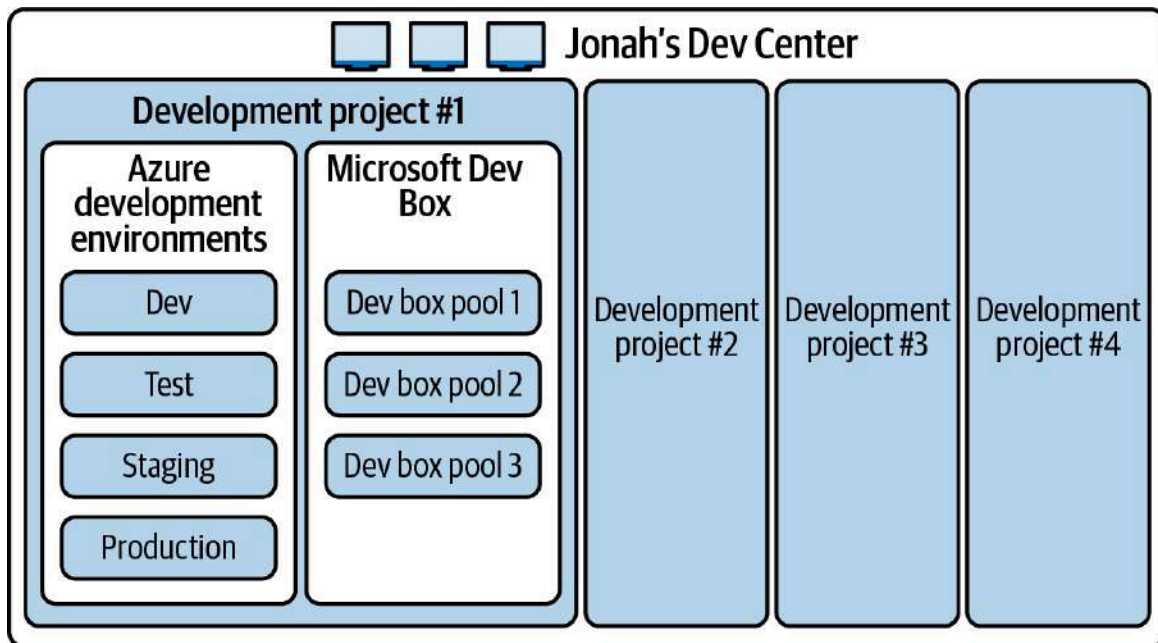


Figure 14-7. Major components of Microsoft Dev Box

Once you have created your Dev Center on Azure, you can set up the essential components, such as the development projects you need in it.

You must set up your environment types in each project, as shown in **Figure 14-8**. A dev box is composed of several dev box tools.

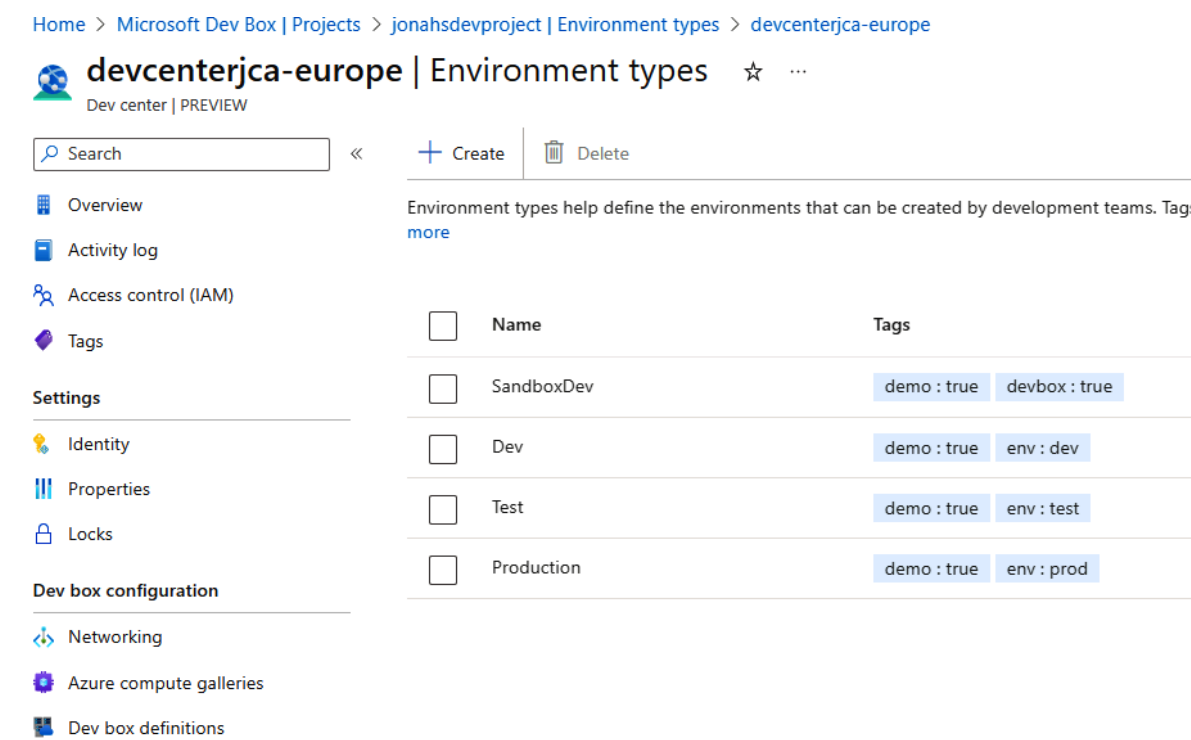


Figure 14-8. Environment types on Azure Portal that you can configure based on Microsoft Dev Box

TIP

A developer can connect to a Dev Box environment through an RDP Client or through the Microsoft Developer Portal, which can be accessed using <https://devportal.microsoft.com>.

These components in your projects are configured with network connectivity, preconfigured tools, security, authentication, and other necessary Azure integrations. They work together to provide developers with a powerful and efficient development environment for building, testing, and deploying applications on Azure.

Azure Deployment Environments in Microsoft Dev Box

Implementing Dev Box for your development teams requires a decision from the organization and project-level team leads. All team projects involved need to agree on the new working method. This is critical because everyone, including the infrastructure and development teams, must be synchronized and prepared to use these development tools.

Azure Deployment Environments enable development teams to swiftly and securely set up app infrastructures with project-oriented templates that ensure consistency and best practices. Deployment Environments are preconfigured Azure resources under specified subscriptions. Azure governs sandbox, testing, staging, and production subscriptions.

Azure Deployment Environments makes cloud infrastructure development, configuration, and management more effortless. Capturing and sharing IaC templates in source control within your team or business for creating on-demand environments standardizes collaboration. For example, dev infra teams may design environment templates to enforce company security policies and match projects to their appropriate Azure subscriptions.

Furthermore, the Deployment Environments also help developers and DevOps or infrastructure teams in project deployment. Preconfiguring environments and integrating with your CI/CD process is familiar.

Developers can provision environments for deployment pipelines. Each feature branch can have a new dev environment to track updates in a prod-like scenario and use the dev environment when creating pull requests (PRs) for better code reviews. PRs are a method for developers to change, review, merge code, and add comments in a Git repository like GitHub, Azure DevOps, Bitbucket, and other source code control repositories.

The infrastructure administrators and developers involved in this creation and configuration processes are shown in **Figure 14-9**.

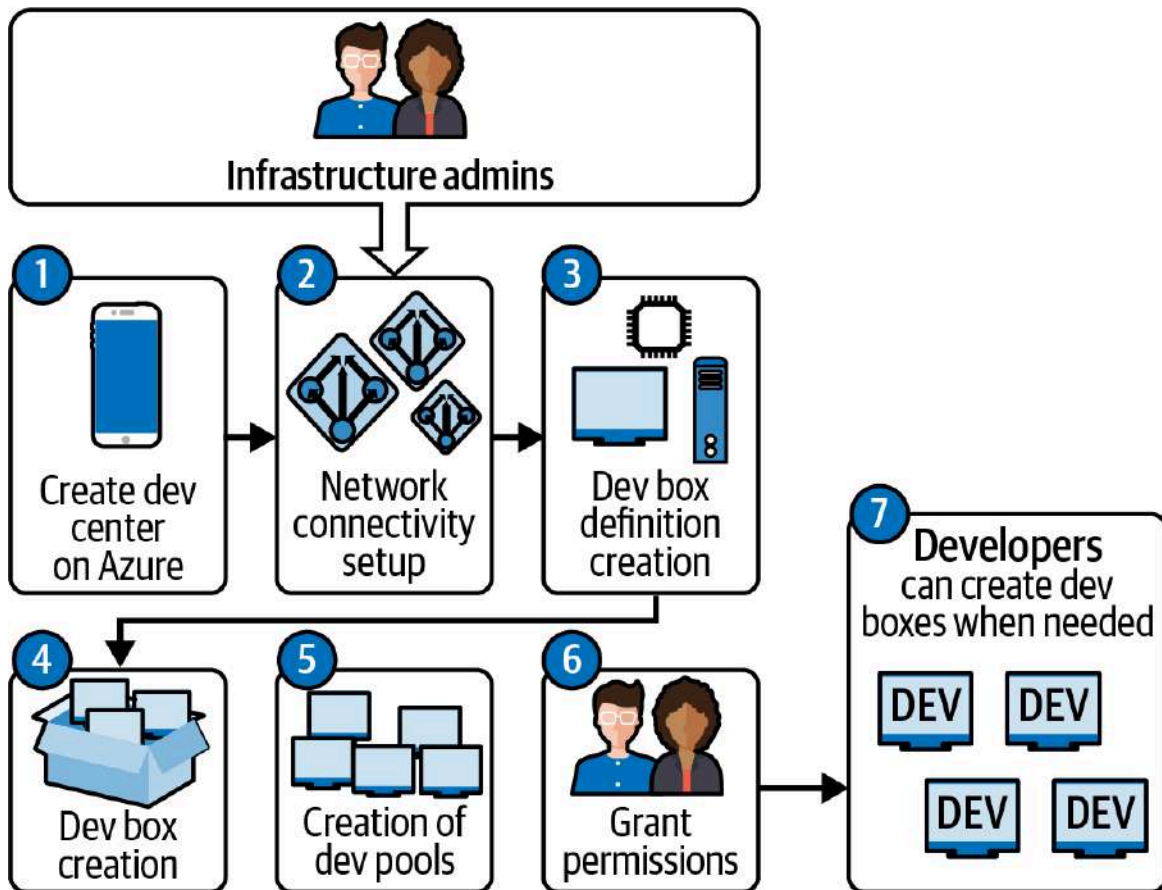


Figure 14-9. Workflow of how infrastructure admins configure a dev box

Once the team agrees to implement Dev Box, infrastructure administrators and developers work together during the initial setup. Most of the steps are configured by an infrastructure admin for the environments, for example, the dev environment.

The configuration can be started based on the following steps:

1. Configure a dev center, which stands in for departments in a business. A development center is a meta-level structure that can be used to manage your dev box's assets. While there is no hard cap on how many development centers a business can have, most have one.
2. Establish an Azure network connection to allow development machines to talk to your internal network. The network bridges the gap between the development center and your company's internal logical networks. Configure your network settings to connect a development

machine to Microsoft Entra ID to do this. Connect only to cloud-based resources with a Microsoft Entra join, or use a hybrid join to connect to both on-premises and cloud-based resources.

3. Next, set up accessible dev boxes. This process is described in the definitions of those boxes. Use a tool that enables a VM image from Azure Marketplace, such as the IDE Visual Studio. As an alternative, developers can create their image and save it to Azure's Compute Gallery.

Note: As the entry point for development groups, dev box projects serve as the hub.

4. In projects, you can use your dev box definitions via dev box pools. You can think of dev box pools as collections of dev box definitions containing comparable parameters. A group of development machines can be shut down at once by setting up an automatic shutdown schedule for the pool.
5. To grant a developer access to the project's dev box pools, give them the Dev Box User role in their DevOps environment.
6. After proper configuration, developers can build and manage their dev boxes through the portal. They can only use the dev box pools linked to the projects where they have been given the Dev Box User status.

Microsoft Dev Box is a handy virtual workstation that provides developers with a preconfigured development environment, tools, and resources for building, testing, and deploying applications on Azure, allowing developers to work more efficiently, accelerate application development, and enhance collaboration.

Azure DevOps and GitHub for Developers

If you have worked with open source repositories, then GitHub is familiar to you as a developer or DevOps engineer. GitHub is a platform that combines open source features. It works well with Azure and other media.

Developer and DevOps tools can increase teamwork, participation, and productivity while addressing enterprise-level security and compliance.

GitHub is integrated with Azure, which accelerates software development by combining the practices and tools that have supported the growth of the largest developer community in the world with Azure products and GitHub services. Azure provides code-to-cloud workflow automation with GitHub Actions for Azure, built-in Visual Studio integration for accelerated Developer Velocity, end-to-end application security with GitHub Advanced Security and Defender for DevOps, and native Azure integrations for faster deployment.

Azure DevOps

Azure DevOps is a cloud-based service for managing the application development lifecycle. It provides services and tools for source control, build and release management, testing, and monitoring. It is helpful for teams developing applications in the cloud that want to automate their development and deployment process with collaboration.

Chapter 11 covered DevOps and infrastructure management in Azure in greater detail. It's important to recall that when developing applications and systems that are prepared for cloud computing or cloud-native, you'll want to leverage the different tools available for CI/CD pipelines, source control, and other DevOps tasks because they help with CI/CD for your applications and collaboration with your teams.

GitHub

As an open source, web-based tool for development, version management, and collaboration, GitHub has become the place for storing repositories for developers and teams worldwide. Version control is one of its primary applications, allowing developers to monitor changes to their code over time and work with others on the same source.

Using GitHub, developers can collaborate in real time on projects, share code, assess each other's work, and seamlessly merge changes.

Additionally, the platform includes a robust issue-tracking system that enables developers to handle bugs, feature requests, and other project-related duties.

CI/CD is supported by GitHub, enabling developers to automate their software development process and ensure code quality. The platform's large and active community of developers who share their code, collaborate on projects, and contribute to open source software is another of its major advantages. Moreover, GitHub provides a free tier for open source repositories, which makes it accessible to engineers of every level of expertise and background.

Integrations are another vital aspect of GitHub, as the platform integrates with many other development services and tools, such as managing projects, CI/CD platforms, and editor platforms. GitHub's various features and benefits make it an indispensable tool for all developers and teams.

GitHub Codespaces

GitHub Codespaces is an interactive development environment that eliminates the need for developers and DevOps engineers to have local development environments by allowing them to write, build, test, and debug code within their web browser. It offers a comprehensive development environment consisting of a code editor, a terminal, a debugger, and other tools required for the software development process.

Developers can use GitHub Codespaces to transition between various projects, interact with others working on the same project, and instantly spin up another development platform for a project without having to set up local development environments.

Furthermore, DevOps engineers, or those working on the infrastructure, can use Codespaces to spin up temporary environments for testing and debugging applications. They can also provide development environments for team members needing access to the requisite hardware or software.

So how does GitHub Codespaces work?

- Whenever a developer creates a Codespace, GitHub builds a cloud-based VM with the required resources (CPU, memory, etc.) to operate their development environment. In this virtual workstation setup, GitHub configures the developer's environment, including their preferred code editor, terminal, and other development tools.
- Once the development environment has been configured, the developer can access it via a web browser by navigating to the Codespace URL.
- The developer can then begin working on their code in the Codespace, just as they would work in a local environment for development. Using the incorporated Git functionality, users can modify their code, test it, troubleshoot it, and then commit to GitHub.

Codespaces also facilitate collaboration, enabling multiple developers to collaborate on the same project within the same Codespace. Developers can grant others access to their Codespace by using GitHub permissions.

When developers have completed working with their code, they can close the Codespace or save it for future use. When developers save their changes to the code, GitHub will maintain the Codespace's state, allowing developers to pick up where they left off.

TIP

Learn more about GitHub Codespaces by watching this [video](#).

Because GitHub Codespaces interacts with various other GitHub tools and workflows, shifting between Codespaces and different GitHub services like pull requests, problems, and code reviews can be done seamlessly. It offers a flexible and convenient solution for programmers and DevOps engineers who prefer to work on code from anywhere without being concerned about setting up and managing their development environments.

GitHub Command-Line Interface

The GitHub command-line interface (CLI) is an open source CLI that brings GitHub to the terminal of your computer. It allows developers to work directly from the command line with problems, pull requests, checks, and releases.

It has sync repository capabilities while working and developing locally on your machine. You should configure a few GitHub CLI configurations depending on your developer preference and your development project standards.

For example, you can use the command `gh repo sync` to synchronize your changes to the destination repository from your source repository. If you don't prefer using the sync, you can also use the `--force` flag post-fix command. You use this command if you prefer to overwrite the desired destination branch with your source repository branch.

Developers who use GitHub CLI save time and the need to switch contexts between the CLI and a web browser. GitHub also provides its developer community with an API that can be used to script virtually any operation and create aliases for every command.

Developers can connect with GitHub repositories and carry out a variety of operations related to GitHub straight from the terminal application installed locally for Windows, macOS, and Linux. This CLI tool supports Github Codespaces.

Command lines commonly used with Github CLI along with Codespaces include:

- `gh codespace code`: Open a Codespace in Visual Studio Code
- `gh codespace create`: Create a Codespace
- `gh codespace delete`: Delete Codespaces based on selection criteria.
- `gh codespace cp`: Copy files between local and remote file systems.

- `gh codespace edit`: Edit a Codespace
- `gh codespace list`: List Codespaces of the authenticated user.
- `gh codespace logs`: View and access logs
- `gh codespace ports`: Change the visibility of the forwarded port
- `gh codespace rebuild`: Rebuild and re-create your Codespace
- `gh codespace ssh`: SSH into a Codespace
- `gh codespace stop`: Stop your GitHub Codespaces

NOTE

With these Github CLI example commands, you have several custom options, such as specifying the name of the Codespace, configuring your organization login, and filtering repositories; you also can use this command to stop a specific user's Codespace.

Unlike Git, the GitHub CLI can authenticate you using your GitHub account's login and password. This adds the benefit of enabling Git to submit changes to your GitHub repositories.

TIP

To install the GitHub CLI, visit the [official installation guide](#) and the GitHub CLI manual for its different uses.

The GitHub CLI provides developers with a time-saving and straightforward method for working with GitHub repositories. It allows developers to carry out everyday GitHub tasks such as building and administering their repositories, managing problems and pull requests, and working with GitHub actions, workflows, and other purposes.

GitHub for Enterprise

GitHub Enterprise is an enterprise-grade version of the GitHub platform that can be customized for businesses with requirements and prerequisites. This enterprise version of GitHub is available only for companies that sign up for the service. It provides increased safety, personalization, and command while maintaining the same robust collaboration and development tools that have made GitHub popular among programmers.

GitHub CLI can be used with GitHub Enterprise just like it is used with the standard GitHub for individual users. However, a few additional steps are required to set up GitHub CLI for use with a GitHub Enterprise instance.

Follow these steps to get started with GitHub CLI for GitHub Enterprise:

1. Install GitHub CLI: Download and install the GitHub CLI tool for your operating system using this GitHub CLI manual: <https://cli.github.com/manual/>. Installation instructions are on the GitHub CLI website.
2. Authenticate with GitHub Enterprise: After installing GitHub CLI, authenticate it with your GitHub Enterprise instance. You can do this by running the command `gh auth login`, which prompts you to enter your GitHub Enterprise credentials.
3. Set the GitHub Enterprise URL: Once you are authenticated, set the URL for your GitHub Enterprise instance using the command `gh config set -h <hostname>`. Replace `<hostname>` with the URL of your GitHub Enterprise instance.
4. Use GitHub CLI commands: With GitHub CLI set up for GitHub Enterprise, you can use the same commands as you would with the public GitHub platform. For example, you can use `gh repo create` to create a new repository, `gh pr create` to create a pull request, or `gh issue list` to list all issues in a warehouse.

GitHub CLI can help developers using the GitHub Enterprise edition to work more efficiently and effectively with their GitHub Enterprise

repositories. It provides a streamlined way to interact with GitHub Enterprise from the command line, enabling you to focus on writing high-quality code and collaborating with other team members.

Learn By Doing (Try It!)

The following are recommended tutorials from Microsoft's official documentation:

- [Create a function app for serverless code execution](#)
- [Get started with Azure CLI](#)
- [Use Visual Studio for modern development](#)
- [Quickstart: Configure Microsoft Dev Box](#)
- [Quickstart for GitHub Codespaces](#)
- [GitHub CLI quickstart](#)
- [Get started using Azure Developer CLI \(azd\)](#)

Summary

Azure provides developers with many powerful tools to build, test, deploy, and manage their applications. These include:

- Command-line interfaces that enable developers to manage Azure resources and automate tasks, including Azure Cloud Shell, Azure CLI, and Azure PowerShell.
- Azure DevOps, a comprehensive suite of tools for managing the entire software development lifecycle, including source control, build and release management, testing, and project management.
- GitHub CLI and GitHub Codespaces, which allow developers to work with GitHub repositories from the command line and a cloud-based

development environment, respectively.

- Microsoft Dev Box, a preconfigured VM that provides a development environment for building and testing Azure applications.

These developer tools for programmers in Azure give a seamless and efficient experience for developing applications in the cloud. In addition to the developer tools designed explicitly for Azure, developer productivity tools and IDEs are essential for efficient and effective software development. Visual Studio and Visual Studio Code (VS Code) are popular IDEs used for Microsoft technologies. They provide developers with powerful features like code completion, debugging, and project management. These IDEs have a vast ecosystem of extensions and plug-ins that can further enhance their capabilities, such as code formatters, linters, and language support for various programming languages.

Developer tools and IDEs can significantly impact developers' productivity and efficiency by automating repetitive tasks, providing code suggestions and auto-completion, and assisting with debugging and testing. Reliable, high-quality tools are essential, especially when working on large-scale projects requiring careful attention to detail and collaboration across teams. The [developer tool list created by Scott Hanselman](#) is a recommended reference, especially for .NET developers working with Azure.

Azure provides developers with a comprehensive set of developer tools tailored to the needs of cloud application development. They play a critical role in software development, ensuring developers can work efficiently and develop high-quality applications.

Check Your Knowledge

1. What everyday tasks can be automated with Azure CLI and PowerShell?
2. What are the benefits of using Microsoft Dev Box?
3. Why would you use GitHub Codespaces?

4. What is the importance of having reliable and efficient developer productivity tools and IDEs?
5. How can extensions and plug-ins in Visual Studio and VS Code enhance developer productivity?

Answers to these questions are in the [Appendix](#).

Recommended Learning Resources

Altaiar, Has, Ingrid Babel, Jack Lee, et al. The Developer's Guide to Azure (Second Edition, ebook). Redmond, WA: Microsoft Press, 2022, <https://oreil.ly/bAupO>.

Andersson, Jonah. "Recommended Learning Materials for .NET + Azure Development + Infrastructure as Code (IaC) + Azure DevOps + DevSecOps + Security in Azure + Developer Inspiration." GitHub repository, <https://oreil.ly/aeOia>.

Azure User Group Sweden. "Azure Developer CLI: "azd up" - One command to rule them all." YouTube video, Oct 21, 2023, <https://oreil.ly/JMami>

Chacon, Scott, and Ben Straub. Pro Git. Berkeley, CA: Apress, 2014, <https://oreil.ly/mCIRM>.

"Get Started with GitHub Documenations." GitHub Docs, https://oreil.ly/HTR_3.

"Github Copilot: Your AI Pair Programmer." GitHub, <https://oreil.ly/adSIk>.

"GitHub Skills." GitHub, <https://oreil.ly/SekcO>.

"How to Run the Azure CLI in a Docker Container." Microsoft Learn, August 8, 2023, <https://oreil.ly/u8Ecl>.

"Key Concepts for Microsoft Dev Box." Microsoft Learn, October 12, 2023, <https://oreil.ly/xQHUP>.

Pollack, Gregg, and Olivier Lacan, “Code School: Git Real.” Pluralsight course, [*https://oreil.ly/jell3*](https://oreil.ly/jell3).

“Visual Studio Product Family Documentation.” Microsoft Learn, [*https://oreil.ly/EhJlY*](https://oreil.ly/EhJlY).

Afterword by Maxim Salnikov

Getting Ready for the Transformation

You've just completed your technical onboarding to cloud computing and Microsoft Azure. Keep this book close at hand on your desktop so you can refer to the relevant chapters, look at the sample architectures, and learn from the code examples again and again. Jonah Andersson, with whom I have had the pleasure and honor of collaborating on multiple community-focused projects, wrote this technical masterpiece to be your day-to-day helper and guide as you explore the endless possibilities of cloud technology.

Based on current technical trends and my more than 20 years of experience in development and technical leadership, let me offer some practical advice after reading Jonah's book. First, while the knowledge you gained is still fresh in your memory and your fingers remember solutions for the "Learn By Doing" hands-on exercises, take the opportunity to get the Azure certifications listed in "Who Should Read This Book." Reading carefully and doing the labs of the chapters relevant to your target certification is a solid foundation for passing the exam. Next, after the Digital Transformation, powered for years by Microsoft Cloud as the industry's most trusted and comprehensive cloud, we are entering the Era of AI with an inevitable AI Transformation. "Every business is a software business" proclaimed technology pioneer Watts S. Humphrey over two decades ago; today, we can refine it to "every business is an AI business." Are you, as a developer, software architect, or IT professional, ready for this era? To answer confidently and positively, pay special attention to **Chapter 6** of this book and continue with a deeper dive into learning (also with "by doing" postfix) the Azure OpenAI services. Today, it's one of the smartest investments you can make in your career.

AI Transformation is not only about what we build but also how we build. My last (but certainly not least) advice is: don't miss the opportunity to boost your developer efficiency by pair-programming with AI-driven copilots. Studies and surveys clearly show that, whether you're writing code, cloud-related scripts, or configurations, you are significantly more productive, satisfied, and happier with an AI copilot. Try it out by pair-programming the "Learn By Doing" labs with GitHub Copilot.

Happy transforming!

Maxim Salnikov

Developer Productivity Business Lead at Microsoft Western Europe

Final Words from the Author

As I sit down to write this book's final chapter, I can't help but reflect on the journey I took to write this book. It involved countless hours of spare time spent researching, learning, and writing my knowledge and the experience I've gained as a developer who has worked with passion and dedication to learn about and develop the topic of this book, Microsoft Azure. The journey of creating this book has been enriching and fulfilling for me.

As a first-time author, I am blessed and grateful to have had the opportunity to write about Azure, and writing this book has taught me so much. I truly appreciate the remarkable efforts, large and small, that others have made for me to finish the book that is in your hands (or on your tablet/phone if you are reading an e-book version). The excellent forewords and afterword for this book, the personal effort invested by each contributor, and you, who picked this book and have invested time to learn about Microsoft Azure—this is what I call community, a community of collaborative learning.

As a teacher, tech trainer, lifelong learner, and developer, I believe in the value of learning by doing, which includes sharing, teaching, and writing.

At its core, this book is about empowering developers, users, and organizations to harness the power of Azure to build and deploy powerful cloud-based applications. In this book, I've provided a comprehensive overview of Azure and how to develop applications with it. Whether you read this book as an IT professional, an IT leader, an IT architect, or an experienced developer, or even if you're someone just starting your journey to the cloud, I hope that this book has provided you with valuable insights and knowledge that you can use to take your Azure development to the next level.

Learning about Azure continues, though. Azure is more than just a collection of services. It's a modern cloud platform that is constantly

evolving, with new features and capabilities always being added. Please continue exploring Azure and stay updated with its latest developments. You can find many of these resources at the end of each chapter to help guide you on your continued learning journey. Although Azure is constantly changing, some things remain constant. The common factors remain, such as the importance of security, scalability, and reliability in our applications.

Throughout this book, I've emphasized the importance of designing, developing, and deploying secure, scalable, and reliable applications. These are recommended when developing in Azure, and I've provided practical advice and guidance in these areas.

I appreciate you taking your precious time to read this book to the end. Writing a book is a labor of passion, and it's always gratifying to know that people find value in what you've written. If this book has helped you, then it is a success. And if it has inspired you to dive deeper into Azure and its cloud services, it is a triumph.

As a next step, consider taking Microsoft exams to obtain Microsoft Azure certifications, for example: Azure Developer Associate (AZ-204), DevOps Engineers Expert (AZ-400), Azure Solutions Architect Expert (AZ-305), Azure Administrator Associate (AZ-104), and other **Microsoft certifications** will enhance your cloud skills within Microsoft technologies including Azure.

Enjoy the opportunity to explore all the incredible possibilities Azure offers. With warmth and sincerity, thank you for taking this ride with me.

I look forward to hearing, reading, and seeing all the fantastic things you'll build with Azure. So, learn more, create something unique and valuable with Microsoft Azure, and share it with the world and your friends!

If you need guidance in your learning journey, feel free to reach out and connect with me so we can learn together!

Sincerely,

Jonah Carrio Andersson, (a.k.a. Jonah Andersson)

Feel free to connect with me through the following channels:

- *Twitter:* [@cjkodare](#)
- *LinkedIn:* [Jonah Andersson](#)
- *Website:* <https://jonahandersson.tech> **Book website:*
<https://learningmicrosoftazure.com>

Appendix A. Check Your Knowledge Answers

Following are the answers for the “Check Your Knowledge” questions at the end of each chapter. I hope that you can see how much you have learned by going back and forth through the chapters as you are learning about Microsoft Azure.

Chapter 1

1. *Cloud computing* is a delivery of computing service like servers, applications, storage, and compute resources, over the internet on an pay-as-you-go basis or any other pricing model that suits any use case.
2. A *public cloud* is a deployment model where the cloud infrastructure is available to the public or any organization and the resources are shared with other organizations or other cloud provider tenants. On the other hand, a *private cloud* is exclusive to one organization where both infrastructure and resources are managed on a private network.
3. The *shared responsibility model* on the cloud is important, especially for cloud security and cloud responsibility. The model provides clear, transparent expectations and responsibilities for the public cloud provider and its users.
4. A *hybrid cloud* is a combination of public and private clouds that is usually implemented in one IT infrastructure and as one entity. On the other hand, *multi-cloud* means using cloud solutions from multiple public cloud providers.
5. *CapEx (capital expenditures)* is spending for the up-front costs of setting up physical IT infrastructure such as costs for data centers, servers, network, technical personnel, backup and maintenance, etc. *OpEx (operational expenditures)* are the costs related to day-to-day operations. A pay-as-you-go subscription for cloud services through a public cloud provider is an example of OpEx.

Chapter 2

1. False. A regional pair should be two regions within the same geography.
2. True.
3. Compute, Storage, Networking, Security, Databases.
4. False. Azure Functions is the service you need if you want to develop event-driven applications in serverless environments.
5. An Azure region is a set of data centers deployed within a latency-defined perimeter and connected in a dedicated regional low-latency network. Azure availability sets are used to protect resources from system failures within an Azure data center.

Chapter 3

1. Azure VM is the virtual machine itself on Azure that you can provision based on your demand while Azure Virtual Machine Scale Sets let you create multiple Azure VMs and manage them with load-balancing features and more.
2. Azure Spot VMs allow users to purchase VMs in Azure from a pool of unused spare capacity at a significantly reduced price, up to 90% cheaper than the pay-as-you-go option.
3. Azure Container Apps gives the flexibility of using serverless containers with the ability to autoscale without managing infrastructure. It also supports tools to simplify the development of containers using DAPR.
4. Azure Durable Functions enables us to write stateful workflows in a serverless infrastructure.
5. The Q# programming language is part of the Quantum Development Kit (QDK) and requires an Azure account to create an Azure Quantum Workspace.

Chapter 4

1. *Azure Virtual Network* or Azure VNet is the fundamental building block for your private network in Azure.
2. Azure Web Application Firewall (WAF), Azure DDoS Protection, Azure Firewall, Azure Private Link, network security groups (NSG).
3. Azure ExpressRoute is an ideal networking service if want to establish a private and secure connection between an on-premises network and Microsoft Azure.
4. **Network security groups.**
5. Azure Content Delivery Network (CDN) is a network service that enables you to distribute content to your users from a central location.

Chapter 5

1. Structured data is highly specific and has a predefined format managed using SQL (Structured Query Language). Non-structured data is usually categorized as qualitative data, which is hard to analyze and process using traditional data methods and tools. Non-structured data is also called NoSQL and does not have a predefined data model.
2. **Azure SQL Managed Instance.**
3. Azure Storage Explorer.
4. Azure Cosmos DB is a powerful, fast, and fully managed NoSQL database for modern app development.
5. Azure Blob Storage is an object storage for large unstructured data. Azure File Storage is a cloud-based file system that is distributed.

Chapter 6

1. Artificial intelligence (AI) applies advanced analysis and logic-based techniques used to interpret events, support and automate decisions, and take action using training data, ML models, and algorithms. Through AI technology, we can train machines or systems to simulate human behavior and to make decisions based on trained models and data.
2. **Fairlearn** is an open source, community-driven project to help data scientists improve fairness of AI systems.
3. Artificial intelligence (AI) is a technology that helps us develop and enable a machine to simulate human behavior while machine learning (ML) is a subset of AI that allows a machine or system to automatically learn from past data without programming explicitly.
4. Azure Cognitive Services for Speech, Languages, Vision, and Decision Making.
5. Because we need AI solutions or products that provide privacy and security. Responsible AI helps reduce unfair biases based on race, gender, nationality, etc.

Chapter 7

1. Even though big data, data analytics, and data science have similarities and differences, they are helpful to organizations, even individuals. They help us handle a massive amount of data of different types and extract critical information from them to make essential and strategic decisions. They are also helpful in finding and identifying new ideas and possibilities.
2. One possible recommendation to solve this security risk is to enhance the cybersecurity practice in your organization as a security strategy for the security of big data tools and initiatives. Also, expand the knowledge of the team working with big data solutions to think about data security to ensure information protection when necessary.
3. A few examples of the known challenges in implementing big data solutions are finding the right people with skills to work with big data, security, and handling the challenge of the growth of big data.
4. Apache Hadoop is an open source framework used in big data analytics. It is efficient in storing and processing large datasets of different types and sizes. It provides the capability to perform parallel clustering on multiple computers for big data analytics.
5. Azure Synapse Analytics does not pull all the data into a data lake and then process it like Data Lake Analytics. Instead, it performs analytics using the logic provided by code and retrieves the data from any Azure-based data sources.

Chapter 8

1. Freestyle answer. Author's point of view: IoT technologies help us in our daily manual tasks. Using IoT devices in our practical tasks saves time.
2. Freestyle answer. Author's point of view: The top two things are how to make IoT technologies more secure and compliant. A third challenge is building practical, useful solutions to our daily tasks or manual routines.
3. Freestyle answer. Author's point of view: Building IoT solutions in the cloud is efficient because of the benefits of autoscaling, scalability, flexibility, and users' ability to fully manage it from anywhere.

Chapter 9

1. MFA is a security feature that forces users to authenticate and identify themselves with more forms of authentication before accessing resources. With Microsoft Entra ID, you can configure MFA for your users to prevent unauthorized access to your organization's resources.
2. Microsoft Defender for Cloud is a cloud-native security solution providing threat protection and management for cloud resources in Azure and cloud providers such as AWS and GCP. This security solution can help detect and respond to malware, phishing attacks, and unauthorized resource access.
3. Best practices for securing keys and secrets in Azure Key Vault include using role-based access control, enabling logging and monitoring, and implementing network security best practices.
4. To ensure compliance with regulatory requirements, you can use Azure Policy to enforce security and compliance requirements, implement continuous monitoring and auditing, and conduct regular security assessments.
5. DevSecOps is a practice that integrates security practices into the DevOps process. It differs from traditional DevOps in prioritizing security throughout the development lifecycle.
6. Azure Firewall is a network security service that provides network-level protection for your Azure virtual networks. It helps protect your virtual networks by filtering traffic based on network and application rules you define.

Chapter 10

1. Integration with cloud services in Azure enables users to access their data and applications from a variety of environments or infrastructure. It helps provide agility, adaptability, and scalability. In addition, it helps companies lower their expenditures and increase the return on investment.
2. Applications with backends suitable for an architecture that implements microservices. Applications that get used as APIs for internal and external consumption can utilize the APIM service, which serves as an API gateway that separates clients from microservices, adds security, and optimizes the development of applications built as microservices by managing cross-cutting concerns like authorization, authentication, throttling, caching, transformation, and monitoring. It includes a self-service developer portal for API discovery, API lifecycle management, and API monitoring as a full-lifecycle API management solution.
3. Integration of microservices, managing REST APIs, maintaining mobile applications, data management for Internet of Things (IoT) devices, and monitoring of website activity are some of the use cases for brokering messaging technology. Messaging broker technology can also manage connections within on-premises systems and cloud-based components in hybrid cloud environments. It provides greater control over interservice communications, ensuring that data is transmitted between app components in a reliable, safe, and efficient way.
4. Azure Web PubSub can be used in developing live dashboards or real-time collaboration tools.
5. Azure Event Grid and Azure Event Hub are messaging services that can be utilized for various use cases. These two services differ in how event data is made available to subscribers. The data that has been

digested is transmitted to subscribers through Event Grid, whereas users can access the data by pulling it from Event Hub.

Chapter 11

1. Azure DevOps is a cloud-based solution that provides a streamlined approach to software development lifecycle management. Its features help teams collaborate and efficiently manage version control, automated builds, deployment, and rollback strategy during automatic landing zone creation. Azure DevOps helps speed up software releases and reduces time-to-market through CI/CD and continuous proactive monitoring.
2. Application Insights also monitors the availability, performance, and usage of web applications hosted on Azure. It leverages the robust data analysis platform in Azure Monitor to provide you with insights into applications. It enables you to detect and diagnose errors and performances issues without waiting for a user to report them.
3. Azure DevTest Labs is a service that enables you to create, use, and manage IaaS, VMs, and PaaS environments in labs. It helps promote efficiency, consistency, and cost control by keeping all resource usage within the lab and classroom context.
4. Azure Policy is a service in Azure that enables you to create, assign, and manage policies. Through these policies, you can enforce rules on your Azure resources to be compliant with organizational or corporate standards and service-level agreements. Azure Policy detects and scans Azure resources to help you identify which ones are not compliant.
5. Azure Network Watcher provides a suite of tools to monitor, diagnose, view metrics, and important logs for Azure IaaS networking resources. It enables you to monitor network health of IaaS products like virtual machines, virtual networks, application gateways, load balancers, etc.

Chapter 12

1. Azure Monitor is a cloud monitoring service that provides visibility into the performance and health of applications, infrastructure, and networks running in Azure. It helps with cloud management by providing insights into various metrics, logs, and alerts to help detect and diagnose issues quickly.
2. Azure Automation can be used for various scenarios such as automated infrastructure management, backup and restore scaling, and provisioning resources. It can also be used for runbook automation, which helps automate repetitive tasks and help streamline operational processes.
3. Azure Resource Manager (ARM) is a service that provides a consistent and secure way to deploy and manage resources in Azure. Using ARM, you can manage your resources into groups and also add tags and categorize them for better organization and management.
4. Azure Blueprints is a service that allows you to automate the deployment of repeatable environments in Azure. It helps with cloud management by providing a standardized way to deploy resources across multiple domains, ensuring consistency and compliance with organizational standards. Blueprints are used to enforce policies and manage access to resources.
5. Some best practices for optimizing cloud costs in Azure include monitoring usage and costs, identifying and eliminating unused resources, resizing resources to match demand, leveraging reserved instances and spot instances, and implementing cost optimization strategies.

Chapter 13

1. The Cloud Adoption Framework for Azure is a comprehensive guide that provides best practices and guidance for organizations to plan, build, and manage cloud solutions using Azure. It helps organizations adopt cloud solutions effectively.
2. Microsoft Assessments help organizations assess their current infrastructure, applications, and business processes to determine cloud adoption and modernization readiness. They provide recommendations and guidance on optimizing applications and infrastructure for Azure, making it easier for organizations to plan and execute their cloud adoption strategy effectively.
3. The Well-Architected Framework for Azure provides best practices and guidance for organizations to design and operate reliable, secure, efficient, and cost-effective cloud solutions in Azure. It provides a framework for organizations to optimize their applications and infrastructure for cloud-native services, making it easier to take advantage of the scalability and flexibility of the cloud.
4. Azure offers a range of hybrid and multi-cloud solutions, including Azure Arc, Azure Stack, Azure Arc-enabled Kubernetes, Azure API Management, Azure ExpressRoute, Azure Site Recovery, and Azure VPN Gateway. These solutions enable organizations to manage resources across different environments and cloud providers, making it easier to build and manage hybrid and multi-cloud environments that are more efficient, secure, and resilient.
5. Azure Arc-enabled Kubernetes allows you to build, deploy and manage Kubernetes clusters across different environments. This includes on-premises, multi-cloud, and edge environments. This helps organizations manage and operate their Kubernetes clusters consistently, regardless of where they are deployed.

Chapter 14

1. Common tasks that can be automated with Azure CLI and PowerShell include creating and configuring Azure resources such as virtual machines, storage accounts, and databases; managing resource groups and permissions; deploying and managing Azure applications; and automating standard maintenance and management tasks.
2. Some benefits of using Microsoft Dev Box include having a preconfigured development environment, managing different environment types, and deploying applications to Azure.
3. GitHub Codespaces provides a convenient way to work on code from anywhere without worrying about setting up and maintaining local development environments. It's beneficial for collaboration and spinning up temporary environments for testing and debugging applications.
4. Having reliable and efficient developer productivity tools and IDEs is critical for ensuring developers can work efficiently and create high-quality applications. These tools can automate repetitive tasks, provide code suggestions and auto-completion, and assist with debugging and testing. They can significantly impact the productivity and efficiency of developers and contribute to better software development.
5. Visual Studio and VSCode have vast ecosystems of extensions and plug-ins that can enhance their capabilities. Examples of these extensions and plug-ins include Copilot, IntelliSense, code formatters, and language support for various programming languages. Developers can use these extensions to tailor their IDE to their specific needs and preferences, making their work more efficient and enjoyable.

Index

A

ACA (Azure Container Apps), [Azure Container Apps-Azure Container Apps](#)

- AKS (Azure Kubernetes Service), [Azure Kubernetes Services](#)
 - CI/CD support, [Azure Kubernetes Services](#)
 - container orchestration, [Azure Kubernetes Services](#)
 - container security, [Azure Kubernetes Services](#)
 - DevOps CI/CD and lifecycle, [Azure Kubernetes Services](#)
 - hybrid containers, [Azure Kubernetes Services](#)

access control management, [Managing and Organizing Resources Using Azure Resource Groups](#)

ACI (Azure Container Instances), [Azure Container Instance](#)

ACR (Azure Container Registry), [Azure Container Registry](#)

AD FS (Active Directory Federation Services), [Active Directory Federation Services](#)

ADC (Application Delivery Controller), [Azure Application Gateway](#)

aggregator pattern, [Aggregator pattern](#)

AI (artificial intelligence), [Artificial Intelligence on Azure: An Introduction-Artificial Intelligence on Azure: An Introduction, Chapter 6](#)

- Azure Applied AI Services, [Azure Applied AI Services-Azure Applied AI Services](#)

- Azure OpenAI Service, [Azure OpenAI Service and Evolution of Chat-GPT](#)
- benefits to business, [Benefits of AI to Businesses-Benefits of AI to Businesses](#)
- CV (computer vision), [AI Technology Innovations and Terms You Need to Know](#)
- Edge AI, [AI Technology Innovations and Terms You Need to Know](#)
- ethics, [Ethical and Responsible AI on Azure-Ethical and Responsible AI on Azure](#)
- generative, [AI Technology Innovations and Terms You Need to Know](#)
- HPC (high-performance computing), [Azure Applied AI Services](#)
- infrastructure, [Azure Applied AI Services](#)
- introduction, [Artificial Intelligence on Azure: An Introduction](#)
- IoT (Internet of Things) with AI, [AI Technology Innovations and Terms You Need to Know](#)
- jobs and careers, [Why Should You Learn AI?](#)
- NLP (natural language processing), [AI Technology Innovations and Terms You Need to Know](#)
- programming languages supported, [Artificial Intelligence on Azure: An Introduction](#)
- reasons to learn, [Why Should You Learn AI?](#)
- regulations, [Artificial Intelligence on Azure: An Introduction](#)
- reinforcement learning, [Machine Learning](#)
- Responsible AI, [Ethical and Responsible AI on Azure](#)

- synthetic data, [AI Technology Innovations and Terms You Need to Know](#)
- virtual agents, [AI Technology Innovations and Terms You Need to Know](#)

AI Builder for Power Platform, [AI Builder for Power Platform](#)

AKS (Azure Kubernetes Service), [Azure Compute for Developing Fully Managed Systems](#), [Azure Kubernetes Services](#)

- ACI (Azure Container Instances), [Azure Container Instance](#)
- ACR (Azure Container Registry), [Azure Container Registry](#)
- CI/CD support, [Azure Kubernetes Services](#)
- container orchestration, [Azure Kubernetes Services](#)
- container security, [Azure Kubernetes Services](#)
- DevOps support, [Azure Kubernetes Services](#)
- hybrid containers, [Azure Kubernetes Services](#)

algorithms, Azure Load Balancer, [Azure Load Balancer](#)

Alibaba Cloud, [Alibaba Cloud](#)

Analysis Services, [Azure Analysis Services](#)

analytics

- big data, [Big Data Use Cases in Azure](#)
- NoSQL, [Azure NoSQL for Big Data and Analytics](#)
- Power BI, embedded, [Power BI Embedded Analytics](#)
- Stream Analytics, [Azure Stream Analytics](#)
- Synapse Analytics, [Azure Synapse Analytics](#)

Android Studio, [Android Studio](#)

any-to-any (IPVPN) connection, [Azure ExpressRoute](#)

Apache Cassandra, [Azure Cosmos DB APIs](#)

Apache Hadoop, [Azure HDInsight for Hadoop](#), R Server, HBase, Spark, and [Storm Clusters](#)

API (application programming interface)

- APIM (Azure API Management), [Azure API Management](#), [Benefits of Azure API Management](#)
 - API gateway, [Azure API gateway](#)
 - API management plane, [Azure API management plane](#)
 - developer portal, [Azure APIM developer portal](#)
- keys for authentication, [Authentication and Security on Azure Maps](#)
- Microsoft Graph API, [Managed Identities on Azure](#)
- vulnerabilities, [Cybersecurity and Why It Matters](#)

API for MongoDB, [Azure Cosmos DB APIs](#)

API gateway, [Azure API gateway](#)

API management lifecycle, [API Management Lifecycle-API Management Lifecycle](#)

API management plane, [Azure API management plane](#)

APIM (Azure API Management), [Azure API Management](#), [Benefits of Azure API Management](#)

- API gateway, [Azure API gateway](#)
- API management plane, [Azure API management plane](#)

- developer portal, [Azure APIM developer portal](#)

App Service, [Azure Compute for Developing Fully Managed Systems](#),
[Azure App Service-Azure Web App for Containers](#)

- application development, [Azure App Service](#)
- autoscaling, [Azure App Service](#)
- fault tolerance, [Azure App Service](#)
- frameworks, [Azure App Service](#)
- high availability, [Azure App Service](#)
- Hybrid Connections, [Azure App Service](#)
- integration, [Azure App Service](#)
- programming languages, [Azure App Service](#)
- support for containers, [Azure Web App for Containers-Azure Web App for Containers](#)

App Service Plan, [Azure App Service](#)

Append Blobs, [Azure Blob Storage](#)

application data input validation, [Application Data Input Validation](#)

application delivery

- Azure Application Gateway, [Azure Application Gateway](#)
- Azure CDN (Content Delivery Network), [Azure CDN](#)
- Azure Front Door, [Azure Front Door](#)
- Azure Traffic Manager, [Azure Traffic Manager](#)

application development

- DevOps and, Modern Application Development and DevOps, Continuous integration-Continuous integration
 - CD (continuous deployment), Continuous deployment-Continuous deployment
 - continuous monitoring, Continuous monitoring
 - continuous testing, Continuous testing-Continuous testing
 - core, The Core of DevOps and Its Function in Application Development

Application Gateway, Azure Load Balancer

Application Insights, Monitoring and Infrastructure Management in Azure
 application-to-application integration, Types of Cloud Integration in Azure
 applications

- Azure VMs, Development and applications-Development and applications
- IoT (Internet of Things), Different Types of IoT Applications
- networking services, Azure Networking Services Categories
 - Azure DDoS Protection, Azure DDoS Protection
 - Azure Firewall, Azure Firewall-Azure Firewall
 - Azure Load Balancer, Azure Load Balancer-Key uses of Azure Load Balancer
 - Azure Private Link, Azure Private Link
 - Azure WAF (Web Application Firewall), Web Application Firewall
 - NSGs (Network Security Groups), Network security group (NSG)

ARM (Azure Resource Manager), [Azure Resource Manager](#), [Managed Identities on Azure](#), [Infrastructure as Code Using Azure Resource Manager and Bicep](#)-[Infrastructure as Code Using Azure Resource Manager and Bicep](#), [Azure Resource Manager-Azure Resource Manager](#)

- [templates](#), [Azure Resource Manager](#)

ARM templates, [Cloud Infrastructure Automation and Management](#)

Async HTTP APIs, [Async HTTP APIs-Async HTTP APIs](#)

authentication

- [Azure Maps](#), [Authentication and Security on Azure Maps](#)
- built-in, [Azure Container Apps](#)
- IAM (Identity and Access Management) service, [Authentication and authorization](#)
- identity providers, [Azure Container Apps](#)
- Managed Identities, [Managed Identities on Azure](#)
- MFA (multi-factor authentication), [Chapter 9](#)

authorization

- IAM (Identity and Access Management) service, [Authentication and authorization](#)

automation, [Azure Automation-Azure Automation](#)

AutoML (automated machine learning), [Automated Machine Learning \(AutoML\)](#)

autoscaling, [Azure Virtual Machine Scale Sets](#), [Scaling Options for Azure VM Scale Sets](#)

- App Service, [Azure App Service](#)

Availability Set, [Azure Resource Groups](#)

availability zones, [Azure Availability Zones](#), [Azure Application Gateway](#)

[AWS \(Amazon Web Services\)](#), [Amazon \(AWS\)](#)

Azure

- as public cloud provider, [Microsoft Azure as a Public Cloud Provider-Features of Azure Portal](#)
- cloud services, [Microsoft Azure](#)
- shared responsibility model, [Shared Responsibility in Cloud Computing and Azure](#)
 - security, [Shared Responsibility Model Offers Cloud Security Advantages](#)

[Azure Advisor](#), [Monitoring and Infrastructure Management in Azure](#), [Cost Management Optimization for Azure](#)

[Azure AI Immersive Reader](#), [Azure Applied AI Services](#)

[Azure AI Metrics Advisor](#), [Azure Applied AI Services](#)

[Azure App Service plan](#), [Azure App Service](#)

[Azure Application Gateway](#), [Azure Application Gateway](#)

[Azure Applied AI Services](#), [Azure Applied AI Services](#)

- [AI Immersive Reader](#), [Azure Applied AI Services](#)
- [AI Metrics Advisor](#), [Azure Applied AI Services](#)
- [Bot Service](#), [Azure Applied AI Services](#)
- [Cognitive Search services](#), [Azure Applied AI Services](#)
- [cognitive services](#), [Azure Applied AI Services](#)
- [databricks](#), [Azure Applied AI Services](#)

- Form Recognizer, [Azure Applied AI Services](#)
- IoT Edge, [Azure Applied AI Services](#)

Azure Arc, [Azure Arc](#)

- Azure Arc-enabled Kubernetes, [Azure Arc-Enabled Kubernetes](#)

Azure Automation, [Azure Automation-Azure Automation](#)

Azure Bastion, [Azure Bastion-Azure Bastion](#), [Azure Network Security](#)

Azure Bicep, [Azure Resource Manager](#), [Infrastructure as Code Using Azure Resource Manager and Bicep](#)

Azure Blob Storage, [Azure Blob Storage-Azure Blob Storage](#)

Azure Blueprints, [Azure Blueprints \(Preview\)](#)

- benefits, [Azure Blueprints \(Preview\)](#)
- creation, [Creation and Deployment of Azure Blueprints-Creation and Deployment of Azure Blueprints](#)
- deployment, [Creation and Deployment of Azure Blueprints-Creation and Deployment of Azure Blueprints](#)
- deprecation plan, [Creation and Deployment of Azure Blueprints](#)
- Zero Trust methodology, [Azure Blueprints for Zero Trust Security and Cloud Migration-Azure Blueprints for Zero Trust Security and Cloud Migration](#)

Azure Budgets, [Cost Management Optimization for Azure](#)

Azure CDN (Content Delivery Network), [Azure CDN](#)

Azure Chaos Studio, [Security perspective: Shifting left versus shifting right](#)

Azure CLI, [Chapter 14](#)

Azure CLI (Command Line Interface), [Azure Command-Line Interface \(CLI\)](#)-[Azure Command-Line Interface \(CLI\)](#)

Azure Cloud Shell, [Azure Cloud Shell](#)-[Azure Cloud Shell](#)

- Predictive IntelliSense, [Predictive IntelliSense in Azure Cloud Shell](#)

Azure Cognitive Services, [Azure AI and Cognitive Services](#)-[Azure AI and Cognitive Services](#)

Azure Compute, [Azure Compute for Developing Fully Managed Systems, Learn by Doing \(Try It!\)](#)

Azure compute services, [Our Journey to the Modern Cloud, Azure Compute for Developing Fully Managed Systems-Learn by Doing \(Try It!\)](#)

Azure Container Instances (ACIs), [Azure Container Instance](#)

Azure Container Registry (ACR), [Azure Container Registry](#)

Azure Cosmos DB, [Azure Cosmos DB](#)-[Azure Cosmos DB](#)

- APIs, [Azure Cosmos DB APIs](#)
 - API for MongoDB, [Azure Cosmos DB APIs](#)
 - Cassandra API, [Azure Cosmos DB APIs](#)
 - Core (SQL) API, [Azure Cosmos DB APIs](#)
 - Gremlin API, [Azure Cosmos DB APIs](#)
 - Table API, [Azure Cosmos DB APIs](#)
- consistency levels, [Azure Cosmos DB consistency levels](#)-[Azure Cosmos DB consistency levels](#)
- global distribution, [Global distribution and replication using Azure Cosmos DB](#)-[Global distribution and replication using Azure Cosmos DB](#)

- replication, [Global distribution and replication using Azure Cosmos DB](#)-[Global distribution and replication using Azure Cosmos DB](#)

Azure Cost Management, [Azure Cost Management Tools-Best Practices for Azure Cost Management](#)

Azure Database Migration Service, [Azure Cosmos DB APIs](#)

Azure DDoS Protection, [Azure DDoS Protection](#)

Azure Deployment Environments, [Azure Deployment Environments in Microsoft Dev Box](#)-[Azure Deployment Environments in Microsoft Dev Box](#)

Azure Developer CLI (azd), [Azure Developer CLI \(azd\)](#)-[Azure Developer CLI \(azd\)](#)

Azure DevTest Labs, [Development and applications](#)

Azure DNS (Domain Name System), [Azure Domain Name System](#)-[Azure Domain Name System](#)

Azure ExpressRoute, [Azure ExpressRoute](#)-[Azure ExpressRoute](#), [Azure ExpressRoute](#)

- ExpressRoute Direct, [Azure ExpressRoute](#)
- Global Reach, [Azure ExpressRoute Global Reach](#)

Azure Files, [Azure Files](#)

Azure Firewall, [Azure Firewall](#)-[Azure Firewall](#), [Azure Network Security](#), [Chapter 4](#)

Azure Firewall Premium, [Azure Firewall](#)

Azure Firewall Standard, [Azure Firewall](#)

Azure for Education, [Community Cloud](#)

Azure Front Door, [Azure Load Balancer](#), [Azure Front Door](#), [Azure Application Gateway](#)

Azure Functions, [Azure Compute for Developing Fully Managed Systems](#), [Azure Functions](#)

- bindings, [Components of Azure Functions](#)
- Durable Functions, [Azure Durable Functions](#)
 - activity functions, [Activity functions-Activity functions](#)
 - client functions, [Client functions-Orchestrator functions](#)
 - entity functions, [Entity functions-Entity functions](#)
 - orchestration patterns, [Fan-out / fan-in](#)
 - orchestrator functions, [Orchestrator functions-Orchestrator functions](#)
 - stateful orchestration patterns, [Application patterns for serverless stateful workflows-Aggregator pattern](#)
- durable functions
 - Orchestrator trigger, [Orchestration Triggers Kickstart Durable Functions-The orchestrator is deterministic](#)
- programming languages, [Components of Azure Functions](#)
- triggers, [Components of Azure Functions](#)

Azure geographies, [Azure Geographies](#)

Azure Government, [Community Cloud](#)

Azure Government Cloud, [Control Results of Azure Maps with Geographic Scope](#)

Azure HDInsight, [What Is Big Data?](#)

- Hadoop and, [Azure HDInsight for Hadoop, R Server, HBase, Spark, and Storm Clusters](#)

Azure Hybrid Benefit, [Scaling Options for Azure VM Scale Sets, Cost Management Optimization for Azure](#)

Azure Hypervisor, [Cloud Hypervisor: The Key to Virtualization in the Cloud](#)

Azure IoT

- actions, [Azure IoT](#)
- insights, [Azure IoT](#)
- things, [Azure IoT](#)

Azure Key Vault, [Azure Key Vault-Azure Key Vault](#)

Azure Kubernetes Services (AKS) (see [AKS \(Azure Kubernetes Service\)](#))

Azure Lab Services, [Community Cloud, Development and applications](#)

Azure Load Balancer, [Azure Load Balancer-Key uses of Azure Load Balancer, Azure Application Gateway](#)

Azure Log Analytics, [Monitoring and Infrastructure Management in Azure](#)

Azure Logic Apps

- actions, [Azure Logic Apps Components](#)
- benefits, [Benefits and Uses of Azure Logic Apps-Benefits and Uses of Azure Logic Apps](#)
- connectors, built-in, [Azure Logic Apps Components](#)
- pricing tiers, [Pricing Tiers of Azure Logic Apps](#)
- triggers, [Azure Logic Apps Components](#)
- workflows, [Azure Logic Apps Components](#)

Azure Machine Learning workspace, [Azure Machine Learning](#)

Azure Managed Disks, [Azure Managed Disks](#)

Azure Management Groups, [Azure Management Groups](#)

Azure Maps

- authentication, [Authentication and Security on Azure Maps](#)
- developing
 - Maps Android SDKs, [Developing using web and mobile software development kits](#)
 - Maps iOS SDK, [Developing using web and mobile software development kits](#)
 - Maps Web SDK, [Developing using web and mobile software development kits](#)
 - REST APIs, [Develop using REST APIs for the Maps Search service](#)
- European API, [Control Results of Azure Maps with Geographic Scope](#)
- Event Grid and, [Maps Integrations with Azure Event Grid](#)
- geographic scope, [Control Results of Azure Maps with Geographic Scope](#)
- geospatial services, [Azure Maps](#)
- interactive maps, [Azure Maps](#)
- location APIs, [Azure Maps](#)
- mobility services, [Azure Maps](#)
- real-time traffic, [Azure Maps](#)
- security, [Authentication and Security on Azure Maps](#)

- spatial operations, [Azure Maps](#)

Azure Marketplace, [Features of Azure Portal](#), [Azure Networking](#), [Azure Resource Manager](#)

Azure MDC (Modular Datacenter), [Azure Space: Networking Beyond the Clouds](#)

Azure mobile app, [Azure Portal](#)

Azure Monitor, [Monitoring and Infrastructure Management in Azure](#), [Azure Monitor for Monitoring and Reliability](#)-[Azure Monitor for Monitoring and Reliability](#), [Cost Management Optimization for Azure](#), Chapter 12

- Application Insights, [Monitoring and Infrastructure Management in Azure](#)

Azure Monitor Network Insights, [Azure Monitor Network Insights](#)

Azure Network Watcher, [Azure Network Watcher](#), [Monitoring and Infrastructure Management in Azure](#)

Azure OpenAI Service, [Azure OpenAI Service and Evolution of Chat-GPT](#)-[Azure OpenAI Service and Evolution of Chat-GPT](#)

Azure Orbital, [Azure Space: Networking Beyond the Clouds](#)

- Azure Orbital Emulator, [Azure Space: Networking Beyond the Clouds](#)

Azure Policy, [Azure Policy for Compliance and Policy Management](#)-[Azure Policy for Compliance and Policy Management](#)

Azure Portal, [Azure Portal](#)

- features, [Features of Azure Portal](#)-[Features of Azure Portal](#)
- services, [Azure Portal](#)

Azure PowerShell, [Azure PowerShell](#)-[Azure PowerShell](#)

Azure Private Link, [Azure Private Link](#), Chapter 4

Azure QDK, [Azure Quantum Development Kit](#)

Azure Quantum, [Azure Quantum](#)

- Azure QDK, [Azure Quantum Development Kit](#)

Azure Queues, [Azure Queue Storage](#)

Azure Region Pairs, [Azure Region Pairs](#)

Azure Regions, [Azure Regions](#)

Azure Repos, [Azure DevOps](#)

Azure Reservations, [Cost Management Optimization for Azure](#)

Azure Resource Manager (ARM) (see [ARM \(Azure Resource Manager\)](#))

Azure SDK for iOS, [Xcode](#)

Azure Service Bus, [Azure Service Bus: Cloud Messaging Broker Service](#)-
[Azure Service Bus: Cloud Messaging Broker Service](#)

- implementations, [Choosing the Right Azure Cloud Messaging Implementation](#)
- namespaces, [Namespaces](#)
- queues, [Queues](#)
- subscriptions, [Topics and subscriptions](#)
- topics, [Topics and subscriptions](#)

Azure Service Bus Queue, [Azure Functions](#)

Azure Site Recovery, [Azure Site Recovery](#)

Azure Space, [Azure Space: Networking Beyond the Clouds](#)

- Azure Orbital, [Azure Space: Networking Beyond the Clouds](#)

- Azure Orbital Emulator, [Azure Space: Networking Beyond the Clouds](#)

Azure Spot VMs, [Scaling Options for Azure VM Scale Sets](#), [Cost Management Optimization for Azure](#)

Azure SQL Database, [Azure SQL as a Fully Managed Database Service](#)

- deployment options, [Azure SQL deployment options-Azure SQL deployment options](#)

Azure Stack, [Azure Stack](#)

Azure Storage, [Azure Storage](#)

- Azure Blob Storage, [Azure Blob Storage-Azure Blob Storage](#)
- data services, [Azure Storage](#)
- database services, [Database Services in Azure](#)

Azure Table Storage, [Azure Table Storage-Azure Table Storage](#)

Azure Testing, [Azure DevOps](#)

Azure Traffic Manager, [Azure Traffic Manager](#)

Azure Virtual Desktop, [Community Cloud](#)

Azure Virtual WAN, [Azure Virtual Wide Area Network-Azure Virtual Wide Area Network](#)

Azure VMs, [Azure Compute for Developing Fully Managed Systems](#), [Azure Virtual Machines and Virtual Machine Scale Sets](#), [Azure Virtual Machines](#), [Chapter 3](#)

- (see also VMs (virtual machines))
- applications, [Development and applications-Development and applications](#)

- development, [Development and applications-Development and applications](#)

Azure VMware, [Azure VMware Solution](#)

Azure VNet, [Azure Virtual Network, Chapter 4](#)

- Azure resources, [Azure Virtual Network](#)
- Internet communication, [Azure Virtual Network](#)
- network traffic filtering, [Azure Virtual Network](#)
- network traffic routing, [Azure Virtual Network](#)
- VNet peering, [Azure VNet Peering-Azure VNet Peering](#)

Azure VPN Gateway, [Azure VPN Gateway](#)

- connections
 - P2S (point-to-site VPN), [Different types of VPN gateway connections](#)
 - S2S (site-to-site VPN), [Different types of VPN gateway connections](#)
 - VNet-to-VNet (IIPsec/IKE VPN tunnel), [Different types of VPN gateway connections](#)

Azure WAF (Web Application Firewall), [Web Application Firewall](#)

Azure Web App for Containers, [Azure Web App for Containers](#)

B

BaaS (backend as a service), [Serverless Computing: Function as a Service and Backend as a Service-Serverless Computing: Function as a Service and Backend as a Service](#)

BI (Business Intelligence)

- descriptive analytics and, [Data Analytics](#)

Bicep, [Cloud Infrastructure Automation and Management, Infrastructure as Code Using Azure Resource Manager and Bicep](#)

big data, [Big Data, Structured Databases, and Non-Structured Databases, Chapter 7](#)

- Analysis Services, [Azure Analysis Services](#)
- architecture, [Identifying big data architecture, Azure Big Data and Analytics Services](#)
- building, [Building, Configuring, and Deploying Big Data on Azure-Identifying big data architecture](#)
- configuring, [Building, Configuring, and Deploying Big Data on Azure-Identifying big data architecture](#)
- Data Factory, [Azure Data Factory](#)
- Data Lake Gen2, [Azure Data Lake Storage](#)
- Data Lake storage, [Azure Data Lake](#)
- databricks, [Azure Databricks](#)
- deploying, [Building, Configuring, and Deploying Big Data on Azure-Identifying big data architecture](#)
- goal evaluation, [Evaluation of a big data goal and solution](#)
- HDInsight, [Azure HDInsight for Hadoop, R Server, HBase, Spark, and Storm Clusters](#)
- managed services, [Azure Big Data and Analytics Services](#)
- Microsoft Purview, [Microsoft Purview for Data Governance](#)
- NoSQL, [Azure NoSQL for Big Data and Analytics](#)

- open source technologies, [Azure Big Data and Analytics Services](#)
- pipelines
 - Data Catalog, [Azure Data Catalog](#)
 - Data Factory, [Azure Data Factory](#)
- Power BI embedded analytics, [Power BI Embedded Analytics](#)
- production environment, [Preparation of Production Environment](#)
- Stream Analytics, [Azure Stream Analytics](#)
- Synapse Analytics, [Azure Synapse Analytics](#)
- use cases
 - analytics, [Big Data Use Cases in Azure](#)
 - data engineering, [Big Data Use Cases in Azure](#)
 - databases, [Big Data Use Cases in Azure](#)
 - ML (machine learning), [Big Data Use Cases in Azure](#)
- variety, [What Is Big Data?](#)
- velocity, [What Is Big Data?](#)
- veracity, [What Is Big Data?](#)
- volume, [What Is Big Data?](#)

bindings, [Components of Azure Functions](#)

BLOB (Binary Large Object) data

- Append Blobs, [Azure Blob Storage](#)
- Block blobs, [Azure Blob Storage](#)
- Page blobs, [Azure Blob Storage](#)

Block blobs, [Azure Blob Storage](#)

Bot Service, [Azure Applied AI Services](#)

built-in authentication, [Azure Container Apps](#)

C

CaaS (containers as a service), [Containers as a Service](#)

CaC (Configuration as Code), [Configuration as Code-Configuration as Code](#)

CAF (Cloud Adoption Framework), [Cloud Adoption and Migration Anti-Patterns](#), [Cloud Adoption Framework for Azure](#), Chapter 13

- 4-S (start small smart steps), [The 4 S's: Start Small Smart Steps](#)
- adopt component, [Cloud Adoption Framework for Azure](#)
- benefits, [Benefits of the Cloud Adoption Framework for Azure](#)
- cloud rationalization, [Cloud Rationalization-The five Rs of rationalization](#)
- DevOps approach, flexible, [Adopt a flexible DevOps approach](#)
- digital estate and cloud migration, [Digital Estate and the Prerequisites of Cloud Migration-Identifying your organization's digital estate](#)
- govern component, [Cloud Adoption Framework for Azure](#)
- hybrid approach, [Using a hybrid approach](#)
- legacy application modernization, [Modernization of Legacy Applications and Traditional Infrastructure-Modernization of Legacy Applications and Traditional Infrastructure](#)
- manage component, [Cloud Adoption Framework for Azure](#)

- migration anti-patterns, [Cloud Adoption and Migration Anti-Patterns-Cloud Adoption and Migration Anti-Patterns](#)
 - WAF (Well-Architected Azure Framework), [The Five Pillars of a Well-Architected Framework for Azure-Benefits of cost optimization of cloud resources](#)
- phased, interactive approach, [Adopt a phased, interactive approach](#)
- plan stage, [Cloud Adoption Framework for Azure](#)
- ready stage, [Cloud Adoption Framework for Azure](#)
- secure stage, [Cloud Adoption Framework for Azure](#)
- security (see Azure Cosmos DB)
- serverless technologies and automation, [Leverage the benefits of serverless technologies and Cloud automation](#)
- strategy stage, [Cloud Adoption Framework for Azure](#)

CAP (Conditional Access Policies), [Conditional Access policies \(CAP\)-Conditional Access policies \(CAP\)](#)

CapEx (capital expenditure), [Capital Expenditures and Operational Expenditures, Chapter 1](#)

Cassandra API, [Azure Cosmos DB APIs](#)

CD (continuous deployment), [Continuous deployment-Continuous deployment](#)

CDN (Content Delivery Network), [Chapter 4](#)

chaos engineering, [Security perspective: Shifting left versus shifting right](#)

ChatGPT (Generative Pre-trained Transformer), [Azure OpenAI Service and Evolution of Chat-GPT](#)

Check Your Knowledge answers, [Chapter 1-Chapter 14](#)

CI (continuous integration), [Continuous integration-Continuous integration](#)

CI/CD (continuous integration/continuous delivery), [Developer Tools, Monitoring, and DevOps Services, Implementing Security Scanning and Checks in Source Code and CI/CD Pipelines](#)

- DevOps and, [Continuous Integration, Deployment, Testing, and Monitoring-Continuous monitoring](#)

class-based durable entities, [Entity functions](#)

Cloud Adoption Framework (CAF) (see [CAF \(Cloud Adoption Framework\)](#))

Cloud Adoption Security Review, [Useful Microsoft Assessments for Cloud Migration](#)

Cloud Adoption Strategy Evaluator, [Useful Microsoft Assessments for Cloud Migration](#)

cloud bursting, [Hybrid Cloud](#)

cloud computing, [What Is Cloud Computing?-What Is Cloud Computing?, Data Storage and Databases in the Cloud, Chapter 1](#)

- benefits, [Benefits of the Cloud in Software Engineering and IT-Benefits of the Cloud in Software Engineering and IT](#)
- business value, [Cloud Computing for IT Companies-Cloud Computing for IT Companies](#)
- CapEx (capital expenditures), [Capital Expenditures and Operational Expenditures](#)
- cybersecurity, importance of, [Importance of Cybersecurity on Cloud Infrastructure-Importance of Cybersecurity on Cloud Infrastructure](#)
- data breaches, [Cybersecurity and Why It Matters](#)
- database services, [Database Services in Azure](#)

- evolution of, **Evolution of Cloud Computing-Cloud computing**
- hypervisor, **Cloud Hypervisor: The Key to Virtualization in the Cloud-Cloud Hypervisor: The Key to Virtualization in the Cloud**
- modern, **Our Journey to the Modern Cloud**
- OpEx (operational expenditures), **Capital Expenditures and Operational Expenditures**
- providers, benefits of, **Benefits of a Cloud Provider**
- service models
 - BaaS (backend as a service), **Serverless Computing: Function as a Service and Backend as a Service-Serverless Computing: Function as a Service and Backend as a Service**
 - CaaS (containers as a service), **Containers as a Service**
 - DaaS (data as a service), **Data as a Service**
 - FaaS (function as a service), **Serverless Computing: Function as a Service and Backend as a Service-Serverless Computing: Function as a Service and Backend as a Service**
 - IaaS (infrastructure as a service), **Infrastructure as a Service**
 - PaaS (platform as a service), **Platform as a Service**
 - SaaS (software as a service), **Software as a Service, Software as a Service**
- shared responsibility model, **Shared Responsibility in Cloud Computing and Azure**
 - security advantages, **Shared Responsibility Model Offers Cloud Security Advantages**

- versus virtualization, [Cloud Computing Versus Virtualization-Cloud Computing Versus Virtualization](#)
- Zero Trust model, [Zero Trust Methodology in the Cloud-Zero Trust Methodology in the Cloud](#)

cloud development

- services, [Developer Tools, Monitoring, and DevOps Services](#)
- tools, [Developer Tools, Monitoring, and DevOps Services](#)
 - (see also development tools)

cloud engineering, DevSecOps

- shift-left, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps, Shift-left: Integrating security practices before production](#)
 - versus shift-right, [Security perspective: Shifting left versus shifting right-Security perspective: Shifting left versus shifting right](#)
- shift-right, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps, Shift-right: Continuous monitoring and improvement in production](#)
 - versus shift-left, [Security perspective: Shifting left versus shifting right-Security perspective: Shifting left versus shifting right](#)

cloud exchange co-location, [Azure ExpressRoute](#)

cloud integration

- application-to-application, [Types of Cloud Integration in Azure](#)
- business agility and, [Business Agility and Better Business Processes](#)
- cloud-to-business, [Types of Cloud Integration in Azure](#)
- cloud-to-cloud, [Types of Cloud Integration in Azure](#)

- cloud-to-mobile, [Types of Cloud Integration in Azure](#)
- combining, [Types of Cloud Integration in Azure](#)
- data integration, [Types of Cloud Integration in Azure](#)
- efficiency and, [Improved Work Efficiency and Cost Savings](#)
- hybrid, [Types of Cloud Integration in Azure](#)
- methods, [Cloud Integration: An Introduction](#)
- on-premises to cloud, [Types of Cloud Integration in Azure](#)
- reliability of applications and, [Reliability and Scalability of Applications-Reliability and Scalability of Applications](#)
- scalability of applications and, [Reliability and Scalability of Applications-Reliability and Scalability of Applications](#)
- Web APIs
 - composite, [Different Types of Web APIs](#)
 - internal, [Different Types of Web APIs](#)
 - partner, [Different Types of Web APIs](#)
 - public, [Different Types of Web APIs](#)

Cloud Journey Tracker, [Useful Microsoft Assessments for Cloud Migration](#)
 cloud management and governance, [Cloud Infrastructure Management and Governance](#)

- ARM (Azure Resource Manager), [Azure Resource Manager-Azure Resource Manager](#)
- cost management, [Cloud Infrastructure Management and Governance](#)
- disaster recovery, [Cloud Infrastructure Management and Governance](#)

- effectiveness, [Cloud Infrastructure Management and Governance](#)
- organizational policies, [Cloud Infrastructure Management and Governance](#)
- regulatory standards, [Cloud Infrastructure Management and Governance](#)
- resource groups, [Managing and Organizing Resources Using Azure Resource Groups](#)
 - access control, [Managing and Organizing Resources Using Azure Resource Groups](#)
 - cost management, [Managing and Organizing Resources Using Azure Resource Groups](#)
 - management task automation, [Managing and Organizing Resources Using Azure Resource Groups](#)
 - monitoring resources, [Managing and Organizing Resources Using Azure Resource Groups](#)
- resource locks, [Azure Resource Locks for Cloud Assets Protection](#)
- risk management, [Cloud Infrastructure Management and Governance](#)
- security and, [Cloud Infrastructure Management and Governance](#)

cloud management and governance and governance

- future of, [The Evolution of Cloud Management and Governance](#)

cloud migration, [Cloud Adoption and Modernization-Check Your Knowledge](#)

cloud migration services, [Cloud Migration and Hybrid + Multi-Cloud Cloud Services](#)

- (see also Cloud Adoption Framework; Well-architected Framework)

cloud migration, digital estates and, [Digital Estate and the Prerequisites of Cloud Migration-Identifying your organization's digital estate](#)

cloud native infrastructure, [Cloud-Native Infrastructure](#)

- characteristics, [Cloud-Native Infrastructure](#)

cloud rationalization, [Cloud Rationalization-The five Rs of rationalization](#)

cloud resources, misconfiguration, [Cybersecurity and Why It Matters](#)

cloud storage, [Data Storage Management in the Cloud](#)

- Azure Files, [Services for Azure Storage](#)
- Azure Managed Disks, [Services for Azure Storage](#)
- Azure Queue Storage, [Services for Azure Storage](#)
- Azure Storage, [Azure Storage](#)
- Azure Table Storage, [Services for Azure Storage](#)
- benefits, [Benefits of Digital Storage in the Cloud](#)
- data retention policy, [Data Storage Management in the Cloud](#)
- non-structured data, [Big Data, Structured Databases, and Non-Structured Databases](#)
- services
 - Azure Blob Storage, [Azure Blob Storage-Azure Blob Storage](#)
 - Azure Files, [Azure Files](#)
 - Azure Queues, [Azure Queue Storage](#)
 - Azure Table Storage, [Azure Table Storage-Azure Table Storage](#)
- structured data, [Big Data, Structured Databases, and Non-Structured Databases](#)

cloud-based data warehouse, [Big Data, Reporting, and Analytics Services in Azure](#)

cloud-to-business integration, [Types of Cloud Integration in Azure](#)

cloud-to-cloud integration, [Types of Cloud Integration in Azure](#)

cloud-to-mobile integration, [Types of Cloud Integration in Azure](#)

cluster computing, [Cluster computing](#)

Cognitive Search services, [Azure Applied AI Services](#)

command line tools

- Azure CLI, [Azure Command-Line Interface \(CLI\)-Azure Command-Line Interface \(CLI\)](#)
- Azure Cloud Shell, [Azure Cloud Shell-Azure Cloud Shell](#)
 - Predictive IntelliSense, [Predictive IntelliSense in Azure Cloud Shell](#)
- Azure Developer CLI (azd), [Azure Developer CLI \(azd\)-Azure Developer CLI \(azd\)](#)
- Azure PowerShell, [Azure PowerShell-Azure PowerShell](#)
- GitHub CLI, [GitHub Command-Line Interface-GitHub for Enterprise](#)

community cloud, [Community Cloud-Community Cloud](#)

composite Web APIs, [Different Types of Web APIs](#)

compute services, [Compute Services in Azure-Compute Services in Azure](#)

- Azure Portal, [Azure Compute for Developing Fully Managed Systems](#)
- serverless, [Serverless Compute Services-Aggregator pattern](#)

computer vision (CV), [AI Technology Innovations and Terms You Need to Know](#)

Conditional Access Policies (CAP), [Conditional Access policies \(CAP\)](#)-
[Conditional Access policies \(CAP\)](#)

Configuration as Code (CAC), [Configuration as Code](#)

Connection Monitor, [Azure Monitor Network Insights](#)
connectivity

- network services, [Azure Networking Services Categories](#)
 - Azure Bastion, [Azure Bastion](#)-[Azure Bastion](#)
 - Azure DNS (Domain Name System), [Azure Domain Name System](#)-[Azure Domain Name System](#)
 - Azure ExpressRoute, [Azure ExpressRoute](#)-[Azure ExpressRoute Global Reach](#)
 - Azure Virtual WAN, [Azure Virtual Wide Area Network](#)-[Azure Virtual Wide Area Network](#)
 - Azure VNet, [Azure Virtual Network](#)-[Azure Virtual Network](#)
 - Azure VPN Gateway, [Azure VPN gateway](#)-[Different types of VPN gateway connections](#)
 - NAT (Network Address Translation), [Azure NAT gateway for virtual networks](#)-[Azure NAT gateway for virtual networks](#)
 - VNet peering, [Azure VNet Peering](#)-[Azure VNet Peering](#)

Container Apps, [Azure Compute for Developing Fully Managed Systems](#),
[Azure Container Apps](#)-[Azure Container Apps](#)

container orchestration, [Azure Kubernetes Services](#)

container services, [Container Services in Azure](#)

- ACA (Azure Container Apps), [Azure Container Apps](#)-[Azure Container Apps](#)

- AKS (Azure Kubernetes Service), [Azure Kubernetes Services-Azure Kubernetes Services](#)
- AKS (Azure Kubernetes Service), [Azure Containers and Azure Kubernetes Service](#)
 - ACI (Azure Container Instances), [Azure Container Instance](#)
 - ACR (Azure Container Registry), [Azure Container Registry](#)

containerization, [Cloud Hypervisor: The Key to Virtualization in the Cloud](#)
containers

- Azure Web App for Containers, [Azure Web App for Containers](#)
- custom containers, [Azure Web App for Containers](#)
- hybrid, [Azure Kubernetes Services](#)
- Microsoft Defender for Containers, [Microsoft Defender for Containers](#)
- multi-container, [Azure Web App for Containers](#)
- security, [Azure Kubernetes Services](#)

containers as a service (CaaS), [Containers as a Service](#)

- (see also ACI, ACR, Azure Container Apps)

continuous monitoring, [Continuous monitoring](#)

continuous testing, [Continuous testing-Continuous testing](#)

Core (SQL) API, [Azure Cosmos DB APIs](#)

cost management, [Cost Management in Microsoft Azure-Cost Management in Microsoft Azure](#), [Managing and Organizing Resources Using Azure Resource Groups](#), [Cost Management Optimization for Azure](#)

- Azure Advisor, [Cost Management Optimization for Azure](#)
- Azure Budgets, [Cost Management Optimization for Azure](#)

- Azure Cost Management, [Azure Cost Management Tools-Best Practices for Azure Cost Management](#)
- Azure Hybrid Benefit, [Cost Management Optimization for Azure](#)
- Azure Monitor, [Cost Management Optimization for Azure](#)
- Azure Reservations, [Cost Management Optimization for Azure](#)
- Azure Spot VMs, [Cost Management Optimization for Azure](#)

costs

- multicloud and, [What Is Multi-Cloud?](#)
- upfront, [Microsoft Azure Helps Organizations Minimize Up-front Costs-Microsoft Azure Helps Organizations Minimize Up-front Costs](#)

CRM (customer relationship management), [Cloud Integration: An Introduction](#)

custom containers, [Azure Web App for Containers](#)

customer relationship management (CRM), [Cloud Integration: An Introduction](#)

customers

- IoT (Internet of Things) and, [The advantages of IoT](#)

CV (computer vision), [AI Technology Innovations and Terms You Need to Know](#)

Cyber Signals, [Importance of Cybersecurity on Cloud Infrastructure](#)

cybersecurity, [Cybersecurity and Why It Matters](#)

- API vulnerabilities, [Cybersecurity and Why It Matters](#)
- cloud computing and, [Importance of Cybersecurity on Cloud Infrastructure-Importance of Cybersecurity on Cloud Infrastructure](#)

- cloud resource misconfiguration, [Cybersecurity and Why It Matters](#)
- collaboration and, [Responsibility for Security Strategies Is a Collaborative Effort-Responsibility for Security Strategies Is a Collaborative Effort](#)
- data breaches, [Cybersecurity and Why It Matters](#)
- DDoS (distributed denial-of-service), [Cybersecurity and Why It Matters](#)
- insider threats, [Cybersecurity and Why It Matters](#)
- malware, [Cybersecurity and Why It Matters](#)
- MitM (man-in-the-middle) attacks, [Cybersecurity and Why It Matters](#)
- Security Pillar, [Cybersecurity, DevSecOps, and Securing Azure Infrastructure](#)
- SQL injection attacks, [Cybersecurity and Why It Matters](#)
- zero-day exploits, [Cybersecurity and Why It Matters](#)

D

DaaS (data as a service), [Data as a Service](#)

- Oracle Cloud, [Oracle Cloud](#)

Dapr (Distributed Application Runtime), [Azure Container Apps](#)

data

- non-structured, [Big Data, Structured Databases, and Non-Structured Databases](#)
- structured, [Big Data, Structured Databases, and Non-Structured Databases](#)

data analytics, [Data Analytics](#)

- descriptive, [Data Analytics](#)
- diagnostic, [Data Analytics](#)
- predictive, [Data Analytics](#)
- prescriptive, [Data Analytics](#)

data as a service (DaaS), [Data as a Service](#)

data breaches, [Cybersecurity and Why It Matters](#)

Data Catalog, [Azure Data Catalog](#)

data centers, [Benefits of the Cloud in Software Engineering and IT](#)

data engineering, big data, [Big Data Use Cases in Azure](#)

Data Factory, [Azure Data Factory](#), [Azure Data Factory](#)

data integration, [Types of Cloud Integration in Azure](#)

Data Lake, [Azure Data Lake](#)

Data Lake Gen2, [Azure Data Lake Storage](#)

data retention policy, [Data Storage Management in the Cloud](#)

database services, [Core Azure Database Services](#), [Database Services in Azure](#)

- Azure Cosmos DB, [Azure Cosmos DB-Global distribution and replication using Azure Cosmos DB](#)
 - APIs, [Azure Cosmos DB APIs-Azure Cosmos DB APIs](#)
 - consistency levels, [Azure Cosmos DB consistency levels-Azure Cosmos DB consistency levels](#)
 - global distribution, [Global distribution and replication using Azure Cosmos DB-Global distribution and replication using Azure Cosmos DB](#)

- replication, [Global distribution and replication using Azure Cosmos DB](#)-Global distribution and replication using Azure Cosmos DB
- Azure SQL Database, [Azure SQL as a Fully Managed Database Service](#)
 - deployment options, [Azure SQL deployment options](#)-Azure SQL deployment options
 - SQL Server Managed Instance, [Azure SQL deployment options](#)
 - SQL Server on Virtual Machines, [Azure SQL deployment options](#)

databases, big data, [Big Data Use Cases in Azure](#)

databricks, [Azure Applied AI Services](#), Azure Databricks

DDoS (distributed denial of service) protection, [Azure DDoS Protection](#), [Cybersecurity and Why It Matters](#), Azure Network Security, Chapter 4

DDoS Rapid Response, [Azure DDoS Protection](#)

deep learning, [Deep Learning in ML](#)

Defender for IoT, [Securing IoT on Azure using Defender for IoT](#)-Securing IoT on Azure using Defender for IoT

deployment models

- community cloud, [Community Cloud](#)-Community Cloud
- hybrid cloud, [Hybrid Cloud](#)
- private cloud, [Private Cloud](#)
- public cloud, [Public Cloud](#)-Advantages of using a public cloud

descriptive data analytics, [Data Analytics](#)

development tools, [Developer Tools](#), [Monitoring](#), and [DevOps Services](#), [Importance of Development Tools for Developer Productivity](#)

- [Android Studio](#), [Android Studio](#)
- [Azure Developer CLI \(azd\)](#), [Azure Developer CLI \(azd\)](#)-[Azure Developer CLI \(azd\)](#)
- [Azure DevOps](#), [Azure DevOps](#)
- [benefits](#), [Importance of Development Tools for Developer Productivity](#)-[Importance of Development Tools for Developer Productivity](#)
- [command line tools](#)
 - [Azure CLI](#), [Azure Command-Line Interface \(CLI\)](#)-[Azure Command-Line Interface \(CLI\)](#)
 - [Azure Cloud Shell](#), [Azure Cloud Shell](#)-[Azure Cloud Shell](#), [Predictive IntelliSense in Azure Cloud Shell](#)
 - [Azure Developer CLI \(azd\)](#), [Azure Developer CLI \(azd\)](#)-[Azure Developer CLI \(azd\)](#)
 - [Azure PowerShell](#), [Azure PowerShell](#)-[Azure PowerShell](#)
 - [GitHub CLI](#), [GitHub Command-Line Interface](#)-[GitHub for Enterprise](#)
- [Eclipse](#), [Eclipse](#)
- [GitHub](#), [GitHub](#)
- [GitHub CLI](#), [GitHub Command-Line Interface](#)-[GitHub Command-Line Interface](#)
 - [GitHub for Enterprise](#) and, [GitHub for Enterprise](#)-[GitHub for Enterprise](#)

- GitHub Codespaces, [GitHub Codespaces-GitHub Codespaces](#)
- IntelliJ IDEA, [IntelliJ IDEA](#)
- JetBrains Rider, [JetBrains Rider](#)
- Microsoft Dev Box, [Microsoft Dev Box-Microsoft Dev Box](#)
 - Azure Deployment Environments, [Azure Deployment Environments in Microsoft Dev Box-Azure Deployment Environments in Microsoft Dev Box](#)
- SDKs (software development kits), [Azure Software Development Kits-Azure Software Development Kits](#)
- Visual Studio, [Visual Studio and Visual Studio Code-Visual Studio and Visual Studio Code](#)
 - Visual Studio Community, [Visual Studio and Visual Studio Code](#)
 - Visual Studio Enterprise, [Visual Studio and Visual Studio Code](#)
 - Visual Studio for Unity, [Visual Studio and Visual Studio Code](#)
 - Visual Studio Professional, [Visual Studio and Visual Studio Code](#)
- Visual Studio Code, [Visual Studio and Visual Studio Code](#)
 - (see also VS Code)
- Xcode, [Xcode](#)

DevOps, [MLOps and DevOps: What's the Difference?-Deep Learning in ML, Azure DevOps](#)

- analytics, built-in, [Azure DevOps](#)
- application development and, [Modern Application Development and DevOps](#)
 - CD (continuous deployment), [Continuous deployment-Continuous deployment](#)

- CI (continuous integration), [Continuous integration-Continuous integration](#)
- continuous monitoring, [Continuous monitoring](#)
- continuous testing, [Continuous testing-Continuous testing](#)
- core, [The Core of DevOps and Its Function in Application Development](#)
- Azure Repos, [Azure DevOps](#)
- Azure Testing, [Azure DevOps](#)
- dashboard, [Azure DevOps](#)
- GitHub and, [Cloud Development and DevOps with GitHub](#)
 - education and, [GitHub for Education](#)
- Microsoft Defender for DevOps, [Microsoft Defender for DevOps](#)
- pipelines, [Azure DevOps](#)
- reporting, [Azure DevOps](#)
- security in, [Adopting Security in DevOps Practices-Adopting Security in DevOps Practices](#)
- suite, [Azure DevOps](#)

DevOps Server, [Azure DevOps Cloud-Based and On-Premises Solutions](#)

DevOps Services, [Developer Tools, Monitoring, and DevOps Services, Azure DevOps Cloud-Based and On-Premises Solutions](#)

DevSecOps, [DevSecOps: Security in Development, DevOps, and Infrastructure-DevSecOps: Security in Development, DevOps, and Infrastructure](#)

- shift-left, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#), [Shift-left: Integrating security practices before production](#)
 - versus shift-right, [Security perspective: Shifting left versus shifting right-Security perspective: Shifting left versus shifting right](#)
- shift-right, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#), [Shift-right: Continuous monitoring and improvement in production](#)
 - versus shift-left, [Security perspective: Shifting left versus shifting right-Security perspective: Shifting left versus shifting right](#)

DevTest Labs, [Azure DevTest Labs for Training, Testing, and Demos](#)-[Azure DevTest Labs for Training, Testing, and Demos](#)

diagnostic data analytics, [Data Analytics](#)

digital estates, cloud migration and, [Digital Estate and the Prerequisites of Cloud Migration](#)-[Identifying your organization's digital estate](#)

Digital Twins, [Digital Twins](#)-[Digital Twins](#)

digitalization, [Benefits of the Cloud in Software Engineering and IT, Our Journey to the Modern Cloud](#)

disaster recovery, [Cloud Infrastructure Management and Governance](#)

Distributed Application Runtime (Dapr), [Azure Container Apps](#)

- (see also [Azure Container Apps](#); [KEDA](#))

distributed computing, [Grid computing](#)

downtime, multicloud and, [What Is Multi-Cloud?](#)

DTFx (Durable Task Framework), [Azure Durable Functions](#)

durable entities, [Entity functions](#)

Durable Functions, [Azure Durable Functions](#)

- activity functions, [Activity functions-Activity functions](#)
- client functions, [Client functions-Orchestrator functions](#)
- entity functions, [Entity functions-Entity functions](#)
- languages, [Azure Durable Functions](#)
- orchestrator functions, [Orchestrator functions-Orchestrator functions](#)
- Orchestrator trigger, [Orchestration Triggers Kickstart Durable Functions-The orchestrator is deterministic](#)
- stateful workflows and design patterns, [Application patterns for serverless stateful workflows](#)
 - aggregator pattern, [Aggregator pattern](#)
 - Async HTTP APIs, [Async HTTP APIs-Async HTTP APIs](#)
 - fan-out/fan-in, [Fan-out / fan-in-Fan-out / fan-in](#)
 - function chaining, [Function chaining-Function chaining](#)
 - human interaction, [Human interaction-Aggregator pattern](#)
 - monitor pattern, [Monitor pattern-Monitor pattern](#)

E

earthquake simulation, [Cluster computing](#)

Eclipse, [Eclipse](#)

Edge AI, [AI Technology Innovations and Terms You Need to Know](#)

edge computing, Azure Cognitive Services, [Azure AI and Cognitive Services](#)

efficiency, cloud integration and, [Improved Work Efficiency and Cost Savings](#)

encryption, TLS/SSL, [Authentication and Security on Azure Maps](#)

endpoints

endpoints, Microsoft Defender for Endpoints, [Microsoft Defender for Endpoint-Microsoft Defender for Endpoint](#)

enterprise resource planning (ERP), [Oracle Cloud, Cloud Integration: An Introduction](#)

entities, durable, [Entity functions](#)

Entra ID, [Authentication and Security on Azure Maps](#)

- (see also Microsoft Entra ID)

EPM, [Oracle Cloud](#)

ERP (enterprise resource planning), [Oracle Cloud, Cloud Integration: An Introduction](#)

error handling, [Taking Error Handling Seriously: Not Just Debugging but Also Security](#)

ethics, AI (artificial intelligence) and, [Ethical and Responsible AI on Azure-Ethical and Responsible AI on Azure](#)

Event Grid, [Azure Event Grid](#)

- Azure Maps and, [Maps Integrations with Azure Event Grid](#)

events, Web SubPub, [Fundamentals of Azure Web PubSub](#)

ExpressRoute Direct, [Azure ExpressRoute](#)

F

FaaS (function as a service), [Serverless Computing: Function as a Service and Backend as a Service](#)-[Serverless Computing: Function as a Service and Backend as a Service](#)

- (see also [Azure Functions](#); [Durable Functions](#))

fan-out/fan-in, [Fan-out / fan-in](#)-[Fan-out / fan-in](#)

fault tolerance, [App Service](#), [Azure App Service](#)

firewalls

- [Azure Firewall](#), [Azure Firewall](#)-[Azure Firewall](#), [Azure Network Security](#)
- [Azure WAF \(Web Application Firewall\)](#), [Web Application Firewall](#)

Form Recognizer, [Azure Applied AI Services](#)

function chaining, [Function chaining](#)-[Function chaining](#)

function-based durable entities, [Entity functions](#)

functions (see [Azure Functions](#))

Fuzzy Search API, [Develop using REST APIs for the Maps Search service](#)

G

GCP (Google Cloud Platform), [Google Cloud Platform](#)

generative AI (GenAI), [AI Technology Innovations and Terms You Need to Know](#)

GitHub, [Cloud Development and DevOps with GitHub](#), [GitHub](#)

- for education, [GitHub for Education](#)

GitHub CLI, [GitHub Command-Line Interface](#)-[GitHub Command-Line Interface](#)

- [GitHub for Enterprise](#) and, [GitHub for Enterprise](#)-[GitHub for Enterprise](#)

GitHub Codespaces, [GitHub Codespaces](#)-[GitHub Codespaces](#)

GitHub Copilot, [Visual Studio](#) and [Visual Studio Code](#)

GitHub for Enterprise, [GitHub for Enterprise](#)-[GitHub for Enterprise](#)

Google Cloud Platform (GCP), [Google Cloud Platform](#)

Gremlin API, [Azure Cosmos DB APIs](#)

- (see also [Azure Cosmos DB](#))

grid computing, [Grid computing](#)

GSaaS (ground station as a service), [Azure Space: Networking Beyond the Clouds](#)

H

Hadoop, [What Is Big Data?](#), [Azure HDInsight for Hadoop](#), [R Server](#), [HBase](#), [Spark](#), and [Storm Clusters](#)

Hashicorp Terraform, [Infrastructure as Code Using Hashicorp Terraform in Azure](#)

HBase, [Azure HDInsight for Hadoop](#), [R Server](#), [HBase](#), [Spark](#), and [Storm Clusters](#)

HCM (Human Capital Management), [Oracle Cloud](#)

high-performance computing (HPC), [Cluster computing](#), [Azure Applied AI Services](#)

HIPAA (Health Insurance Portability and Accountability Act), [Community Cloud](#)

HPC (high-performance computing), [Cluster computing](#), [Azure Applied AI Services](#)

human interaction, [Human interaction-Aggregator pattern](#)

hybrid cloud, [Hybrid Cloud](#), [Chapter 1](#)

- Azure Arc and
 - Azure Arc-enabled Kubernetes, [Azure Arc-Enabled Kubernetes](#)
- Azure ExpressRoute and, [Azure ExpressRoute](#)
- Azure Site Recovery and, [Azure Site Recovery](#)
- Azure Stack and, [Azure Stack](#)
- Azure VMware and, [Azure VMware Solution](#)
- Azure VPN Gateway and, [Azure VPN Gateway](#)
- cloud bursting, [Hybrid Cloud](#)
- versus multicloud, [Hybrid Cloud Versus Multi-Cloud-Hybrid Cloud Versus Multi-Cloud](#)
- services, [Cloud Migration and Hybrid + Multi-Cloud Cloud Services](#)

hybrid cloud integration, [Types of Cloud Integration in Azure](#)

Hybrid Connections, [Azure App Service](#)

hybrid containers, [Azure Kubernetes Services](#)

hybrid multi-cloud

- Azure Arc and, [Azure Arc](#)

hypervisor, [Cloud Hypervisor: The Key to Virtualization in the Cloud](#)-
[Cloud Hypervisor: The Key to Virtualization in the Cloud](#)

- (see also [Azure Hypervisor](#))

I

IaaS (infrastructure as a service), [Infrastructure as a Service](#)

- Oracle Cloud, [Oracle Cloud](#)

IaC (infrastructure as code), [Benefits of the Cloud in Software Engineering and IT](#), [Developer Tools](#), [Monitoring](#), and [DevOps Services](#), [Infrastructure as Code](#)-[Infrastructure as Code](#)

- ARM (Azure Resource Manager), [Infrastructure as Code Using Azure Resource Manager and Bicep](#)-[Infrastructure as Code Using Azure Resource Manager and Bicep](#)
- Bicep, [Infrastructure as Code Using Azure Resource Manager and Bicep](#)
- Hashicorp Terraform, [Infrastructure as Code Using Hashicorp Terraform in Azure](#)
- Terraform/Bicep/ARM comparison, [When to Consider Azure Terraform over Azure Bicep or ARM](#)-[When to Consider Azure Terraform over Azure Bicep or ARM](#)

IAM (Identity and Access Management) service

- AD FS (Active Directory Federation Services), [Active Directory Federation Services](#)
- authentication, [Authentication and authorization](#)
- authorization, [Authentication and authorization](#)
- Azure Key Vault, [Azure Key Vault](#)-[Azure Key Vault](#)

- Azure RBAC (role-based access control), **Authentication and authorization**
- CAP (Conditional Access Policies), **Conditional Access policies (CAP)-Conditional Access policies (CAP)**
- Identity Protection, **Microsoft Entra ID Protection**
- MFA (multi-factor authentication), **Multi-Factor Authentication**
- Microsoft Defender for Cloud, **Microsoft Defender for Cloud**
 - Defender for App Service, **Microsoft Defender for App Service**
 - Defender for Containers, **Microsoft Defender for Containers**
 - Defender for DevOps, **Microsoft Defender for DevOps**
 - Defender for Endpoints, **Microsoft Defender for Endpoint-Microsoft Defender for Endpoint**
- Microsoft Entra ID
 - hybrid identities, **Hybrid identities on Microsoft Entra ID**
 - Managed Identities, **Managed Identities on Azure-Managed Identities on Azure**
 - PHS (password hash synchronization), **Password hash synchronization**
 - PTA (pass-through authentication), **Pass-through authentication**
 - risk detection, **Microsoft Entra risk detection**
 - security features, **Microsoft Entra ID security features**
- Microsoft Entra ID Connect, **Microsoft Entra Connect for hybrid SSO and authentication**
- Microsoft Sentinel, **Microsoft Sentinel**

- RBAC (role-based access control), [Azure role-based access control \(RBAC\)](#)
- VNet (virtual network), [Azure Network Security](#)
 - Azure Bastion, [Azure Network Security](#)
 - Azure Firewall, [Azure Network Security](#)
 - DDoS protection, [Azure Network Security](#)
 - NSGs (Network Security Groups), [Azure Network Security](#)

Identity and Access Management (see IAM (Identity and Access Management))

identity management services, [Identity Management and Security Services-Identity Management and Security Services](#)

Identity Protection, [Microsoft Entra ID Protection](#)

identity providers, [Azure Container Apps](#)

IDEs (integrated development environments), [Importance of Development Tools for Developer Productivity](#)

- Android Studio, [Android Studio](#)
- Eclipse, [Eclipse](#)
- IntelliJ IDEA, [IntelliJ IDEA](#)
- JetBrains Rider, [JetBrains Rider](#)
- SDKs (software development kits), [Azure Software Development Kits-Azure Software Development Kits](#)
- Visual Studio, [Visual Studio and Visual Studio Code-Visual Studio and Visual Studio Code](#)
 - Visual Studio Community, [Visual Studio and Visual Studio Code](#)

- Visual Studio Enterprise, [Visual Studio and Visual Studio Code](#)
- Visual Studio for Unity, [Visual Studio and Visual Studio Code](#)
- Visual Studio Professional, [Visual Studio and Visual Studio Code](#)
- Visual Studio Code, [Visual Studio and Visual Studio Code](#)
- Xcode, [Xcode](#)

infrastructure as code (IaC), [Benefits of the Cloud in Software Engineering and IT, Infrastructure as Code-When to Consider Azure Terraform over Azure Bicep or ARM](#)

infrastructure, cloud native (see cloud native infrastructure)

insider threats, [Cybersecurity and Why It Matters](#)

integrated development environments (IDEs), [Importance of Development Tools for Developer Productivity](#)

integration, [Chapter 10](#)

- App Service, [Azure App Service](#)

integration platform as a service (iPaaS), [Types of Cloud Integration in Azure](#)

IntelliJ IDEA, [IntelliJ IDEA](#)

interactive voice response (IRV), [AI Technology Innovations and Terms You Need to Know](#)

internal Web APIs, [Different Types of Web APIs](#)

Internet of Agriculture (IoA), [Internet of Things](#)

Internet of Buildings (IoB), [Internet of Things](#)

Internet of Things (IoT) (see IoT (Internet of Things))

Internet of Vehicles (IoV), [Internet of Things](#)

IoA (Internet of Agriculture), [Internet of Things](#)

IoB (Internet of Buildings), [Internet of Things](#)

IoT (Internet of Things), [Oracle Cloud](#), [AI Technology Innovations and Terms You Need to Know](#)

- (see also [Azure IoT](#))
- advantages, [The advantages of IoT-The advantages of IoT](#)
- Azure Digital Twins, [Digital Twins-Digital Twins](#)
- Azure IoT (see [Azure IoT](#))
- complexity, [The disadvantages of IoT](#)
- customers, [The advantages of IoT](#)
- data collection, [The advantages of IoT](#)
- definition, [Internet of Things](#)
- devices, [Internet of Things](#)
 - network analyzers, [Securing IoT on Azure using Defender for IoT](#)
- disadvantages, [The disadvantages of IoT-The disadvantages of IoT](#)
- employee monitoring, [The advantages of IoT](#)
- flexibility, [The disadvantages of IoT](#)
- legality and compliance, [The disadvantages of IoT](#)
- Microsoft Defender for IoT, [Securing IoT on Azure using Defender for IoT-Securing IoT on Azure using Defender for IoT](#)
- privacy, [The disadvantages of IoT](#)
- resource sustainability, [The advantages of IoT](#)
- security, [The disadvantages of IoT](#)

- smart farms, [The advantages of IoT](#)
- solutions
 - applications, [Different Types of IoT Applications](#)
 - components, [Components of an IoT Solution-Components of an IoT Solution](#)
- streams, [The advantages of IoT](#)
- technology, [Making Sense of IoT Technology-Making Sense of IoT Technology](#)
- technology innovation, [The advantages of IoT](#)
- warehouse, [The advantages of IoT](#)
- waste reduction, [The advantages of IoT](#)

IoT Central, [Azure IoT Central](#)

IoT DevKit, [Azure IoT DevKit and Azure-accredited IoT devices](#)

IoT Edge, [Azure Applied AI Services](#)

IoT Hub, [Azure IoT Hub and its device provisioning service](#)

IoT Plug and Play, [Azure IoT Plug and Play app](#)

IoV (Internet of Vehicles), [Internet of Things](#)

iPaaS (integration platform as a service), [Types of Cloud Integration in Azure](#)

IT, cloud computing benefits, [Benefits of the Cloud in Software Engineering and IT-Benefits of the Cloud in Software Engineering and IT](#)

IVR (interactive voice response), [AI Technology Innovations and Terms You Need to Know](#)

J

JetBrains Rider, [JetBrains Rider](#)

K

Kubernetes Event-Driven Autoscaling (KEDA), [Azure Container Apps](#)

- (see also [Azure Container Apps](#))

Kubernetes, Azure Arc-enabled, [Azure Arc-Enabled Kubernetes](#)

L

Landing Zone Review, [Useful Microsoft Assessments for Cloud Migration](#)
languages

- App Service, [Azure App Service](#)
- Azure Functions, [Components of Azure Functions](#)
- Durable Functions and, [Azure Durable Functions](#)

LANs (local area networks), [Cluster computing](#)

- (see also [networking services](#))

legacy applications, modernization, [Modernization of Legacy Applications and Traditional Infrastructure-Modernization of Legacy Applications and Traditional Infrastructure](#)

- (see also [cloud migration](#))

lift-and-shift cloud migration, [Azure Virtual Machines and Virtual Machine Scale Sets](#)

load balancing, [Azure Load Balancer](#)

- (see also [Azure Front Door](#))

local area networks (LANs), [Cluster computing](#)

logic-based techniques, [Artificial Intelligence on Azure: An Introduction](#)

M

mainframe computing, [Mainframe computing](#)

- (see also cloud security; cybersecurity)

malware, [Cybersecurity and Why It Matters](#)

- (see also cybersecurity)

Managed Identities, [Managed Identities on Azure](#)

- (see also IAM (Identity and Access Management))
- PAT (Personal Access Tokens), [Managed Identities on Azure](#)
- Service Principals, [Managed Identities on Azure](#)
- system-assigned, [Managed Identities on Azure](#)
- user-assigned, [Managed Identities on Azure](#)

Map Services (see Azure Maps)

Maps Android SDKs, [Developing using web and mobile software development kits](#)

Maps iOS SDK, [Developing using web and mobile software development kits](#)

Maps Power BI Visual, [Azure Maps](#)

Maps Web SDK, [Developing using web and mobile software development kits](#)

MFA (multi-factor authentication), [Multi-Factor Authentication](#), [Chapter 9](#)

- (see also IAM (Identity and Access Management))

Microsoft Assessments, [Microsoft Assessments for Evaluation and Review](#)

- Cloud Adoption Security Review, [Useful Microsoft Assessments for Cloud Migration](#)
- Cloud Adoption Strategy Evaluator, [Useful Microsoft Assessments for Cloud Migration](#)
- Cloud Journey Tracker, [Useful Microsoft Assessments for Cloud Migration](#)
- Landing Zone Review, [Useful Microsoft Assessments for Cloud Migration](#)
- SMART (Strategic Migration Assessment and Readiness Tool), [Useful Microsoft Assessments for Cloud Migration](#)
- Well-Architected Review for Azure, [Useful Microsoft Assessments for Cloud Migration](#)

Microsoft Azure (see Azure)

Microsoft Cloud for Sovereignty, [Azure Portal](#)

Microsoft Defender for Azure App Service, [Microsoft Defender for App Service](#)

Microsoft Defender for Cloud, [Microsoft Defender for Cloud](#)

- Defender for App Service, [Microsoft Defender for App Service](#)
- Defender for Containers, [Microsoft Defender for Containers](#)
- Defender for DevOps, [Microsoft Defender for DevOps](#)
- Defender for Endpoints, [Microsoft Defender for Endpoint-Microsoft Defender for Endpoint](#)

Microsoft Defender for Containers, [Microsoft Defender for Containers](#)

Microsoft Defender for DevOps, [Microsoft Defender for DevOps](#)

Microsoft Defender for Endpoints, [Microsoft Defender for Endpoint-Microsoft Defender for Endpoint](#)

Microsoft Dev Box, [Microsoft Dev Box-Microsoft Dev Box](#)

- Azure Deployment Environments, [Azure Deployment Environments in Microsoft Dev Box-Azure Deployment Environments in Microsoft Dev Box](#)

Microsoft Entra ID, [User Identities, Roles, and Active Directories in Azure, Azure App Service](#)

- IAM and
 - CAP (Conditional Access Policies), [Conditional Access policies \(CAP\)-Conditional Access policies \(CAP\)](#)
 - hybrid identities, [Hybrid identities on Microsoft Entra ID](#)
 - Managed Identities, [Managed Identities on Azure-Managed Identities on Azure](#)
 - MFA (multi-factor authentication), [Multi-Factor Authentication](#)
 - PHS (password hash synchronization), [Password hash synchronization](#)
 - PTA (pass-through authentication), [Pass-through authentication](#)
 - risk detection, [Microsoft Entra risk detection](#)
 - security features, [Microsoft Entra ID security features](#)

Microsoft Entra ID Connect, [Microsoft Entra Connect for hybrid SSO and authentication](#)

Microsoft Graph API, [Managed Identities on Azure](#)

Microsoft Learn, [Azure Resource Manager](#)

Microsoft Purview, data Governance and, [Microsoft Purview for Data Governance](#)

Microsoft Security, [Importance of Cybersecurity on Cloud Infrastructure](#)

Microsoft Sentinel, [Microsoft Sentinel](#)

Microsoft Zero Trust, [Networking Services in Azure](#)

migration anti-patterns, [Cloud Adoption and Migration Anti-Patterns-Cloud Adoption and Migration Anti-Patterns](#)

- WAF (Well-Architected Framework), [The Five Pillars of a Well-Architected Framework for Azure](#)
 - cost optimization, [Pillar #5: Cost Optimization-Benefits of cost optimization of cloud resources](#)
 - operational excellence, [Pillar #1: Operational Excellence-Automation of processes](#)
 - performance efficiency, [Pillar #4: Performance Efficiency-Pillar #4: Performance Efficiency](#)
 - reliability, [Pillar #3: Reliability-Change management](#)
 - security, [Pillar #2: Security](#)

migration, lift-and-shift, [Azure Virtual Machines and Virtual Machine Scale Sets](#)

MitM (man-in-the-middle) attacks, [Cybersecurity and Why It Matters](#)

ML (machine learning), [Artificial Intelligence on Azure: An Introduction, Azure Machine Learning](#)

- Azure Machine Learning workspace, [Azure Machine Learning](#)
- big data, [Big Data Use Cases in Azure](#)
- deep learning, [Deep Learning in ML](#)

- ML Studio, [Machine Learning Studio](#)
 - AI Builder, [AI Builder for Power Platform](#)
 - AutoML, [Automated Machine Learning \(AutoML\)](#)
 - uses, [Machine Learning Studio](#)
- supervised learning, [Machine Learning](#)
- unsupervised learning, [Machine Learning](#)

ML Studio, [Machine Learning Studio](#)

- AI Builder, [AI Builder for Power Platform](#)
- AutoML, [Automated Machine Learning \(AutoML\)](#)
- uses, [Machine Learning Studio](#)

MLOps (ML operations), [MLOps and DevOps: What's the Difference?](#)-
[Deep Learning in ML](#)

modernization, [Benefits of the Cloud in Software Engineering and IT, Our Journey to the Modern Cloud](#)

modernizing legacy applications, [Modernization of Legacy Applications and Traditional Infrastructure-Modernization of Legacy Applications and Traditional Infrastructure](#)

monitor pattern, [Monitor pattern-Monitor pattern](#)

monitoring

- Azure Advisor, [Monitoring and Infrastructure Management in Azure](#)
- Azure Log Analytics, [Monitoring and Infrastructure Management in Azure](#)
- Azure Monitor, [Monitoring and Infrastructure Management in Azure](#)

- Application Insights, [Monitoring and Infrastructure Management in Azure](#)
- Azure Network Watcher, [Monitoring and Infrastructure Management in Azure](#)

multi-container apps, [Azure Web App for Containers](#)

multi-factor authentication (MFA), [Multi-Factor Authentication](#)

multicloud, [What Is Multi-Cloud?](#)

- versus hybrid cloud, [Hybrid Cloud Versus Multi-Cloud-Hybrid Cloud Versus Multi-Cloud](#)
- services, [Cloud Migration and Hybrid + Multi-Cloud Cloud Services](#)
- solutions, [Multi-Cloud and Hybrid Solutions in Azure](#)

MXCHIP DevKit, [Azure IoT DevKit and Azure-accredited IoT devices](#)

N

NAT (Network Address Translation), [Azure NAT gateway for virtual networks-Azure NAT gateway for virtual networks](#)

natural language processing (NLP), [AI Technology Innovations and Terms You Need to Know](#)

network security, [Azure Network Security](#)

- Azure Bastion, [Azure Network Security](#)
- Azure Firewall, [Azure Network Security](#)
- DDoS protection, [Azure Network Security](#)
- Microsoft Defender for Cloud, [Microsoft Defender for Cloud](#)
 - Defender for App Service, [Microsoft Defender for App Service](#)

- Defender for Containers, [Microsoft Defender for Containers](#)
- Defender for DevOps, [Microsoft Defender for DevOps](#)
- Defender for Endpoints, [Microsoft Defender for Endpoint-Microsoft Defender for Endpoint](#)
- Microsoft Sentinel, [Microsoft Sentinel](#)
- NSGs (Network Security Groups), [Azure Network Security](#)

Network Security Groups (NSGs), [Azure Bastion](#), [Network security group \(NSG\)](#), [Authentication and Security on Azure Maps](#), [Azure Network Security](#), [Chapter 4](#)

networking, [Azure Networking](#)

networking services, [Networking Services in Azure](#)

- application delivery, [Azure Networking Services Categories](#)
 - [Azure Application Gateway](#), [Azure Application Gateway](#)
 - [Azure CDN \(Content Delivery Network\)](#), [Azure CDN](#)
 - [Azure Front Door](#), [Azure Front Door](#)
 - [Azure Traffic Manager](#), [Azure Traffic Manager](#)
- application protection, [Azure Networking Services Categories](#)
 - [Azure DDoS Protection](#), [Azure DDoS Protection](#)
 - [Azure Firewall](#), [Azure Firewall-Azure Firewall](#)
 - [Azure Load Balancer](#), [Azure Load Balancer-Key uses of Azure Load Balancer](#)
 - [Azure Private Link](#), [Azure Private Link](#)
 - [Azure WAF \(Web Application Firewall\)](#), [Web Application Firewall](#)

- NSGs (Network Security Group), [Network security group \(NSG\)](#)
- connectivity, [Azure Networking Services Categories](#)
 - Azure Bastion, [Azure Bastion-Azure Bastion](#)
 - Azure DNS (Domain Name System), [Azure Domain Name System-Azure Domain Name System](#)
 - Azure ExpressRoute, [Azure ExpressRoute-Azure ExpressRoute Global Reach](#)
 - Azure NAT (Network Address Translation), [Azure NAT gateway for virtual networks-Azure NAT gateway for virtual networks](#)
 - Azure Virtual WAN, [Azure Virtual Wide Area Network-Azure Virtual Wide Area Network](#)
 - Azure VNet, [Azure Virtual Network-Azure Virtual Network](#)
 - Azure VPN Gateway, [Azure VPN gateway-Different types of VPN gateway connections](#)
 - VNet peering, [Azure VNet Peering-Azure VNet Peering](#)
- monitoring, [Azure Networking Services Categories](#)
 - Azure Monitor Network Insights, [Azure Monitor Network Insights](#)
 - Azure Network Watcher, [Azure Network Watcher](#)
- security, Microsoft Zero Trust, [Networking Services in Azure](#)

NFS (Network File System), [Services for Azure Storage](#), [Azure Files](#)

NLP (natural language processing), [AI Technology Innovations and Terms You Need to Know](#)

nodes, [Cluster computing](#)

non-structured data, [Big Data](#), [Structured Databases](#), and [Non-Structured Databases](#)

NoSQL, [Azure NoSQL for Big Data and Analytics](#)

NSGs (Network Security Groups), [Azure Bastion](#), [Network security group \(NSG\)](#), [Authentication and Security on Azure Maps](#), [Azure Network Security](#), [Chapter 4](#)

O

OAuth 2.0, [Authentication and Security on Azure Maps](#)

observability features, [Benefits of Azure API Management](#)

on-premises to cloud integration, [Types of Cloud Integration in Azure](#)

OpEx (operational expenditures), [Capital Expenditures and Operational Expenditures](#)

- (see also [Azure Application Insights](#); [Azure IoT](#); [Azure Monitor](#); [cloud computing](#))

Oracle Cloud, [Oracle Cloud](#)

- EPM, [Oracle Cloud](#)
- ERP (enterprise resource planning), [Oracle Cloud](#)
- HCM (Human Capital Management), [Oracle Cloud](#)
- IoT (Internet of Things), [Oracle Cloud](#)
- Oracle Cloud CX, [Oracle Cloud](#)
- Supply Chain Management, [Oracle Cloud](#)

Oracle Cloud CX, [Oracle Cloud](#)

OSI (Open Systems Interconnection) model, [Key uses of Azure Load Balancer](#)

P

PaaS (platform as a service), [Platform as a Service](#)

- Digital Twins, [Digital Twins](#)

PaC (policy as code), [Policy as Code-Policy as Code](#)

packet capturing, [Azure Network Watcher](#)

Page Blobs, [Azure Blob Storage](#)

partner Web APIs, [Different Types of Web APIs](#)

pass-through authentication (PTA), [Pass-through authentication](#)

password hash synchronization (PHS), [Password hash synchronization](#)

Personal Access Tokens (PAT), [Managed Identities on Azure](#)

PHS (password hash synchronization), [Password hash synchronization pipelines](#)

- Data Catalog, [Azure Data Catalog](#)
- Data Factory, [Azure Data Factory](#)

point-to-point Ethernet connection, [Azure ExpressRoute](#)

policy as code (PAC), [Policy as Code-Policy as Code](#)

policy management, [Azure Policy for Compliance and Policy Management-Azure Policy for Compliance and Policy Management](#)

port scanning, Azure Bastion, [Azure Bastion](#)

Power BI, [Power BI Embedded Analytics](#)

PowerShell, [Chapter 14](#)

predictive data analytics, [Data Analytics](#)

Predictive IntelliSense, [Predictive IntelliSense in Azure Cloud Shell](#)

prescriptive data analytics, [Data Analytics](#)

pricing calculator, [Cost Management in Microsoft Azure](#)

- (see also Azure Cost Management)

private cloud, [Private Cloud](#)

PTA (pass-through authentication), [Pass-through authentication](#)

public cloud, [Public Cloud-Advantages of using a public cloud](#), Chapter 1

- providers
 - Alibaba Cloud, [Alibaba Cloud](#)
 - AWS, [Amazon \(AWS\)](#)
 - Azure as, [Microsoft Azure as a Public Cloud Provider-Features of Azure Portal](#)
 - GCP (Google Cloud Platform), [Google Cloud Platform](#)
 - Microsoft Azure, [Microsoft Azure](#)
 - Oracle Cloud, [Oracle Cloud](#)

public Web APIs, [Different Types of Web APIs](#)

Q

quantum computing

- Azure Quantum, [Azure Quantum](#)
 - Azure QDK, [Azure Quantum Development Kit](#)

R

R Server, [Azure HDInsight for Hadoop](#), R Server, HBase, Spark, and Storm Clusters

ransomware, [Cybersecurity and Why It Matters](#)

- (see also cloud security; cybersecurity)

RBAC (role-based access control), [Azure Management Groups](#), [Azure Role-Based Access Control](#), [Azure role-based access control \(RBAC\)](#)

- ARM (Azure Resource Manager) and, [Azure Resource Manager](#)
- Azure Maps, [Authentication and Security on Azure Maps](#)
- IAM and, [Authentication and authorization](#)
- roles, [Azure roles](#)
 - definition, [Azure roles](#)
 - scope, [Azure roles](#)
 - security principal, [Azure roles](#)

reliability of applications, cloud integration and, [Reliability and Scalability of Applications-Reliability and Scalability of Applications](#)

repetitive tasks, [Azure Automation](#)

resource groups, [Azure Resource Groups](#), [Managing and Organizing Resources Using Azure Resource Groups](#)

- access control management, [Managing and Organizing Resources Using Azure Resource Groups](#)
- cost management, [Managing and Organizing Resources Using Azure Resource Groups](#)

- management task automation, [Managing and Organizing Resources Using Azure Resource Groups](#)
- resource monitoring, [Managing and Organizing Resources Using Azure Resource Groups](#)

resource locks, [Azure Resource Locks for Cloud Assets Protection](#)

resources, [Azure Resources-Azure Resources](#)

- ARM (Azure Resource Manager), [Azure Resource Manager](#)
- Azure Monitor, [Azure Monitor for Monitoring and Reliability-Azure Monitor for Monitoring and Reliability](#)

Responsible AI, [Ethical and Responsible AI on Azure](#)

REST API

- ARM (Azure Resource Manager) and, [Azure Resource Manager](#)
- Azure Maps developing, [Develop using REST APIs for the Maps Search service](#)

ring-based deployment, [Security perspective: Shifting left versus shifting right](#)

role-based access controls (RBAC), [Azure role-based access control \(RBAC\)](#)

roles, [Azure roles](#)

- definition, [Azure roles](#)
- scope, [Azure roles](#)
- security principal, [Azure roles](#)
 - (see also managed identities; service principal)

root management groups, [Azure Management Groups](#)

- (see also Azure Resource Groups)

S

SaaS (software as a service), [Software as a Service-Software as a Service](#)

- Oracle Cloud, [Oracle Cloud](#)
- virtual agents, [AI Technology Innovations and Terms You Need to Know](#)

scalability

- application, cloud integration and, [Reliability and Scalability of Applications-Reliability and Scalability of Applications](#)
- Azure Storage, [Azure Storage](#)

scaling, Azure VM Scale Sets, [Scaling Options for Azure VM Scale Sets](#)

SDKs (software development kits), [Azure Software Development Kits-Azure Software Development Kits](#)

SDLC (software development lifecycle), [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#)

Search Address API, [Develop using REST APIs for the Maps Search service](#)

Search Address Reverse Cross Street API, [Develop using REST APIs for the Maps Search service](#)

Secure Virtual Hub, [Azure Firewall](#)

security, [What Is Multi-Cloud?](#), [Cybersecurity and Why It Matters](#)

- (see also Azure Security; cybersecurity; Microsoft Defender for Cloud)
- API vulnerabilities, [Cybersecurity and Why It Matters](#)
- application data input validation, [Application Data Input Validation](#)

- authentication, built-in, [Azure Container Apps](#)
- Azure Maps, [Authentication and Security on Azure Maps](#)
- best practices, [Security Best Practices for Azure-DevSecOps: Security in Development, DevOps, and Infrastructure](#)
- CI/CD pipelines, [Implementing Security Scanning and Checks in Source Code and CI/CD Pipelines](#)
- cloud adoption, [Modernization of Legacy Applications and Traditional Infrastructure](#)
- cloud management and governance and, [Cloud Infrastructure Management and Governance](#)
- cloud resource misconfiguration, [Cybersecurity and Why It Matters](#)
- collaboration and, [Responsibility for Security Strategies Is a Collaborative Effort-Responsibility for Security Strategies Is a Collaborative Effort](#)
- communication, [Secure Communication and Integration Between Applications and APIs](#)
- containers, [Azure Kubernetes Services](#)
- data breaches, [Cybersecurity and Why It Matters](#)
- DDoS (distributed denial-of-service), [Cybersecurity and Why It Matters](#)
- DevOps practices, [Adopting Security in DevOps Practices-Adopting Security in DevOps Practices](#)
- DevSecOps, [DevSecOps: Security in Development, DevOps, and Infrastructure-DevSecOps: Security in Development, DevOps, and Infrastructure](#)

- error handling, [Taking Error Handling Seriously: Not Just Debugging but Also Security](#)
- insider threats, [Cybersecurity and Why It Matters](#)
- integration between applications and APIs, [Secure Communication and Integration Between Applications and APIs](#)
- IoT (Internet of Things), [The disadvantages of IoT](#)
- malware, [Cybersecurity and Why It Matters](#)
 - (see also cybersecurity)
- MitM (man-in-the-middle) attacks, [Cybersecurity and Why It Matters](#)
- multicloud and, [What Is Multi-Cloud?](#)
- network, Microsoft Zero Trust, [Networking Services in Azure](#)
- RBAC (role-based access control), [Azure roles](#)
- security scanning, [Implementing Security Scanning and Checks in Source Code and CI/CD Pipelines](#)
- SQL injection attacks, [Cybersecurity and Why It Matters](#)
- storage best practices, [Azure Storage Security Best Practice Tips](#)
- zero-day exploits, [Cybersecurity and Why It Matters](#)

security data orchestration automated response (SOAR), [Microsoft Sentinel](#)
 Security Information and Event Management (SIEM), [Identity Management and Security Services](#)

security services, [Identity Management and Security Services-Identity Management and Security Services](#)

SEIM (Security Information and Event Management), [Identity Management and Security Services](#)

serverless compute services, [Serverless Compute Services-Aggregator pattern](#)

serverless computing, [Serverless Computing: Function as a Service and Backend as a Service-Serverless Computing: Function as a Service and Backend as a Service](#)

services, [Microsoft Azure Services-Microsoft Azure Services](#)

- categories, [Microsoft Azure Services](#)
- cloud migration, [Cloud Migration and Hybrid + Multi-Cloud Cloud Services](#)
- compute services, [Compute Services in Azure-Compute Services in Azure](#)
- core services, [Overview of Azure Core Services](#)
- database services, [Core Azure Database Services](#)
- DevOps Services, [Developer Tools, Monitoring, and DevOps Services](#)
- hybrid + multicloud, [Cloud Migration and Hybrid + Multi-Cloud Cloud Services](#)
- identity management, [Identity Management and Security Services-Identity Management and Security Services](#)
- network services, [Networking Services in Azure](#)
- security services, [Identity Management and Security Services-Identity Management and Security Services](#)
- storage services, [Core Azure Storage Services](#)

shared responsibility, [Shared Responsibility in Cloud Computing and Azure, Chapter 1](#)

- security, [Shared Responsibility Model Offers Cloud Security Advantages](#)

shift-left approach, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#), [Shift-left: Integrating security practices before production](#)

- versus shift-right, [Security perspective: Shifting left versus shifting right](#)-[Security perspective: Shifting left versus shifting right](#)

shift-right approach, [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#), [Shift-right: Continuous monitoring and improvement in production](#)

- versus shift-left, [Security perspective: Shifting left versus shifting right](#)-[Security perspective: Shifting left versus shifting right](#)

SIEM (Security Information and Event Management), [Microsoft Sentinel](#)
single-page application (SPA), [Azure App Service](#)

SMART (Strategic Migration Assessment and Readiness Tool), [Useful Microsoft Assessments for Cloud Migration](#)

SMB (Server Message Block), [Services for Azure Storage](#), [Azure Files](#)

SNAT (source network address translation), [Azure NAT gateway for virtual networks](#), [Azure NAT gateway for virtual networks](#)

SOAR (security orchestration automated response), [Microsoft Sentinel](#)
software development kits (SDKs), [Azure Software Development Kits](#)-[Azure Software Development Kits](#)

software development life cycle (SDLC), [Cloud Engineering Practices: Shift-Left or Shift-Right in DevSecOps](#)

SPA (single-page application), [Azure App Service](#)

Spark, [Azure HDInsight for Hadoop, R Server, HBase, Spark, and Storm Clusters](#)

Spot VMs, [Chapter 3](#)

- (see also Azure Virtual Machines)

SQL (Structured Query Language)

- injection attacks, [Cybersecurity and Why It Matters](#)

SQL Managed Instance, [Azure SQL deployment options](#)

SQL Server on VMs (virtual machines), [Azure SQL deployment options](#)

stateful workflows, [Application patterns for serverless stateful workflows](#)

- aggregator pattern, [Aggregator pattern](#)
- Async HTTP APIs, [Async HTTP APIs-Async HTTP APIs](#)
- fan-out/fan-in, [Fan-out / fan-in-Fan-out / fan-in](#)
- function chaining, [Function chaining-Function chaining](#)
- human interaction, [Human interaction-Aggregator pattern](#)
- monitor pattern, [Monitor pattern-Monitor pattern](#)

Static Web Apps, [Static Web Apps-Key Benefits and Uses of Azure Static Web Apps](#)

storage

- cloud (see cloud storage)
- Data Lake Gen2, [Azure Data Lake Storage](#)

Storage Explorer, [Chapter 5](#)

storage services, [Core Azure Storage Services](#)

Storm Clusters, [Azure HDInsight for Hadoop, R Server, HBase, Spark, and Storm Clusters](#)

Stream Analytics, [Azure Stream Analytics](#), [Azure Service Bus: Cloud Messaging Broker Service](#)

structured data, [Big Data](#), [Structured Databases](#), and [Non-Structured Databases](#), [Chapter 5](#)

subscriptions, [Azure Subscriptions](#)

Supply Chain Management, [Oracle Cloud](#)

Synapse Analytics, [Azure Synapse Analytics](#)

synthetic data, [AI Technology Innovations and Terms You Need to Know](#)

T

Table API, [Azure Cosmos DB APIs](#)

tasks, repetitive, [Azure Automation](#)

TCO (total cost of ownership)

- calculator, [Cost Management in Microsoft Azure](#)

technology, IoT (Internet of Things), [The advantages of IoT](#)

TensorFlow, [Azure Machine Learning](#)

Terraform, [Cloud Infrastructure Automation and Management](#), [Infrastructure as Code Using Hashicorp Terraform in Azure](#)

Traffic Manager, [Azure Load Balancer](#)

triggers, [Components of Azure Functions](#)

troubleshooting, tools, [Developer Tools](#), [Monitoring](#), and [DevOps Services](#)

U

upfront costs for cloud computing, [Microsoft Azure Helps Organizations Minimize Up-front Costs](#)

- (see also CapEx; OpEx)

V

version control, [Azure DevOps](#)

- (see also Azure DevOps; GitHub)

virtual agents, [AI Technology Innovations and Terms You Need to Know](#)

virtual machines (VM) (see VMs (virtual machines))

virtual networks, [Azure VNet Peering](#)

- Azure Virtual Network, [Chapter 4](#)

Virtual WANs, [Azure Firewall](#)

virtualization, [Cloud Computing Versus Virtualization-Cloud Computing Versus Virtualization](#)

Visual Studio, [Visual Studio and Visual Studio Code-Visual Studio and Visual Studio Code](#)

- Visual Studio Community, [Visual Studio and Visual Studio Code](#)
- Visual Studio Enterprise, [Visual Studio and Visual Studio Code](#)
- Visual Studio for Unity, [Visual Studio and Visual Studio Code](#)
- Visual Studio Professional, [Visual Studio and Visual Studio Code](#)

Visual Studio Code, [Visual Studio and Visual Studio Code](#)

VM Scale Sets, [Azure Compute for Developing Fully Managed Systems, Azure Virtual Machines and Virtual Machine Scale Sets](#)

- autoscaling, [Azure Virtual Machine Scale Sets](#), [Scaling Options for Azure VM Scale Sets](#)
- scaling options, [Scaling Options for Azure VM Scale Sets](#)

VMs (virtual machines), [Cloud Hypervisor: The Key to Virtualization in the Cloud](#), [Azure Compute for Developing Fully Managed Systems](#), Chapter 3

- (see also Azure VMs)
- Managed Identities, [Managed Identities on Azure](#)
- Spot VMs, [Chapter 3](#)
- SQL Server, [Azure SQL deployment options](#)

VNet (virtual network), [Azure Network Security](#)

- Azure Bastion, [Azure Network Security](#)
- Azure Firewall, [Azure Network Security](#)
- DDoS protection, [Azure Network Security](#)
- NSGs (Network Security Groups), [Azure Network Security](#)

VNet peering, [Azure VNet Peering-Azure VNet Peering](#)

- global VNet peering, [Azure VNet Peering](#)

W

WAF (Web Application Firewall), [Chapter 4](#)

WAF (Well-Architected Framework), [The Five Pillars of a Well-Architected Framework for Azure](#)

- cost optimization, [Pillar #5: Cost Optimization-Benefits of cost optimization of cloud resources](#)

- operational excellence, [Pillar #1: Operational Excellence-Automation of processes](#)
- performance efficiency, [Pillar #4: Performance Efficiency-Pillar #4: Performance Efficiency](#)
- reliability, [Pillar #3: Reliability-Change management](#)
- security, [Pillar #2: Security](#)
- Well-Architected Review for Azure, [Azure Well-Architected Review-Benefits of Azure Well-Architected Review](#)

Web APIs

- composite, [Different Types of Web APIs](#)
- internal, [Different Types of Web APIs](#)
- partner, [Different Types of Web APIs](#)
- public, [Different Types of Web APIs](#)

Web SubPub in Azure, [Azure Web PubSub](#)

- architecture, [The Architecture Pattern Used in Azure Web PubSub-The Architecture Pattern Used in Azure Web PubSub](#)
- client events, [Fundamentals of Azure Web PubSub](#)
- event handlers, [Fundamentals of Azure Web PubSub](#)
- event listeners, [Fundamentals of Azure Web PubSub](#)
- groups, [Fundamentals of Azure Web PubSub](#)
- hub, [Fundamentals of Azure Web PubSub](#)
- messages, [Fundamentals of Azure Web PubSub](#)
- server, [Fundamentals of Azure Web PubSub](#)

- users, [Fundamentals of Azure Web PubSub](#)
- uses, [Benefits of Azure Web PubSub](#)
- WebSockets, [Azure Web PubSub](#)
- workflow, [Typical Azure Web PubSub Workflow](#)

WebSockets, [Azure Web PubSub](#)

Well-Architected Framework (WAF) (see WAF (Well-Architected Framework))

Well-Architected Review for Azure, [Azure Well-Architected Review-Benefits of Azure Well-Architected Review](#), [Useful Microsoft Assessments for Cloud Migration](#)

workflows

- Azure Logic Apps, [Azure Logic Apps Components](#)
- Durable Functions and, [Azure Durable Functions](#)

X

Xcode, [Xcode](#)

Z

Zero Trust methodology, cloud computing

- Azure Blueprints, [Azure Blueprints for Zero Trust Security and Cloud Migration-Azure Blueprints for Zero Trust Security and Cloud Migration](#)

Zero Trust model, cloud computing, [Zero Trust Methodology in the Cloud-Zero Trust Methodology in the Cloud](#)

zero-day exploits, [Cybersecurity and Why It Matters](#)

- Azure Bastion, **Azure Bastion**

About the Author

Jonah Carrio Andersson, a.k.a. Jonah Andersson, is a Filipina-Swedish **Microsoft Most Valuable Professional (MVP) for Azure**, Microsoft Certified Trainer, software engineer, DevOps engineer, tech mentor, international public speaker, and an inspiring tech community leader based in Sweden.

Jonah has many years of experience in different tech roles and industries, including software development in C# .NET and cloud development with Microsoft Azure. She is a Microsoft Certified Azure Developer and DevOps Engineer Expert, and holds other certifications. She is also a Microsoft Certified Trainer (MCT) who teaches Microsoft Certifications related to Azure. Jonah also started her own company, Jonah Andersson Tech AB, which provides content and training about cloud development within Microsoft Azure. She holds a BSc. in Computer Science and Agile System Development in Microsoft .NET and Java.

Jonah likes continuous learning; she particularly enjoys solving challenging, complex problems and programming backend. She also likes to connect with other developers and share her technical knowledge at her workplace and through public speaking.

She is the founder of **Azure User Group Sweden**, a global Azure Users Community in Sweden that is an open and inclusive tech community for cloud development in Azure. Jonah organizes the technical sessions related to Microsoft Azure cloud services and is active in sharing her knowledge through this group.

Jonah's mission is to inspire others. She is the co-host of the podcast, **Extend Women in Tech**, and wants to make a difference in the tech world by being a role model. In particular, she advocates for gender equality, diversity, and inclusion in tech and is a mentor to youth and young women, inspiring them to learn about tech and cloud technologies while helping them choose a career path in software engineering. She is the recipient of the Developer of the Year at the Nordic Women in Tech Awards 2023.

If you want to directly reach out to Jonah Andersson, please use the following contact information:

Website: [*https://www.jonahandersson.tech*](https://www.jonahandersson.tech)

LinkedIn: [*https://www.linkedin.com/in/jonahandersson/*](https://www.linkedin.com/in/jonahandersson/)

Twitter: [*https://www.twitter.com/cjkodare*](https://www.twitter.com/cjkodare)

GitHub: [*https://github.com/jonahandersson*](https://github.com/jonahandersson)

Colophon

The animal on the cover of *Learning Microsoft Azure* is a hyacinth macaw (*Anodorhynchus hyacinthinus*). They are extremely intelligent birds that are capable of imitating human speech, and they generally live between 30 and 50 years in the wild and over 50 years in captivity.

Hyacinth macaws are the largest species of macaw. They have an average length of 40 inches and a wingspan of roughly 5 feet. However, they only weigh about as much as a guinea pig (42 ounces). They are predominantly covered in cobalt blue feathers that become more violet on their wings. Their eyes and black beak are surrounded by bright yellow bare skin. Macaws have zygodactyl feet, meaning each foot has two claws facing forward and two claws facing backward. This makes them excellent at perching on branches, climbing trees, and holding food.

Hyacinth macaws can be found in three isolated groups in Central South America, Bolivia, and northern Paraguay. Unlike parrots that enjoy living in tropical rainforest habitats, hyacinth macaws prefer living on the edge of tropical, moist lowland forests, palm savannas, flooded grasslands, and open, dry woodlands with gallery forests.

Through the morning and later afternoon, hyacinth macaws can be found foraging for food. They enjoy eating a variety of nuts from palm trees, fruit, nectar, and seeds. In Brazil and Bolivia, they are an essential part of seed dispersal for 18 different plants.

Unfortunately, hyacinth macaws are considered a vulnerable species with a decreasing population. Their main threats include illegal pet trade, hunting, and habitat destruction. Many of the animals on O'Reilly covers are endangered; all of them are important to the world.

The color cover illustration is by Karen Montgomery, based on an antique line engraving from a loose plate, source unknown. The cover fonts are Gilroy Semibold and Guardian Sans. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.