About the Exam
Exam 70-534 focuses on the skills and knowledge needed to design effective Microsoft Azure public and hybrid cloud solutions.

About Microsoft Certification
Passing this exam earns you a Microsoft Specialist certification in Microsoft Azure, demonstrating your expertise with the Microsoft Azure enterprise-grade cloud platform.

You can earn this certification by passing Exam 70-532, Developing Microsoft Azure Solutions; or Exam 70-533, Implementing Microsoft Azure Infrastructure Solutions; or Exam 70-534, Architecting Microsoft Azure Solutions.

See full details at: microsoft.com/learning

About the Authors
Haishi Bai, Senior Technical Evangelist at Microsoft, focuses on the Microsoft Azure compute platform, including IaaS, PaaS, networking, and scalable computing services.

Steve Maier, Senior Technical Evangelist at Microsoft, specializes in Microsoft Azure.

Dan Stolts, Senior Technical Evangelist at Microsoft, is a technology expert proficient in datacenter technologies.

Prepare for Microsoft Exam 70-534—and help demonstrate your real-world mastery of Microsoft Azure solution design and architecture. Designed for experienced IT pros ready to advance their status, Exam Ref focuses on the critical-thinking and decision-making acumen needed for success at the Microsoft Specialist level.

Focus on the expertise measured by these objectives:
- Describe Microsoft Azure infrastructure and networking
- Help secure resources
- Design an application storage and data access strategy
- Design an advanced application
- Design websites
- Design a management, monitoring, and business continuity strategy

This Microsoft Exam Ref:
- Organizes its coverage by exam objectives
- Features strategic, what-if scenarios to challenge you
- Assumes you have experience designing Microsoft Azure cloud or hybrid solutions and supporting application life cycle management

Architecting Microsoft Azure Solutions

Exam Ref
70-534

microsoft.com/mspress

ISBN 978-0-7356-9744-7

U.S.A. $39.99
Canada $45.99

Recommended

Microsoft Press

Haishi Bai
Steve Maier
Dan Stolts
Contents at a glance

Microsoft certifications xv
Preparing for the exam xviii

CHAPTER 1 Design Microsoft Azure infrastructure and networking 1
CHAPTER 2 Secure resources 63
CHAPTER 3 Design an application storage and data access strategy 129
CHAPTER 4 Design an advanced application 189
CHAPTER 5 Design Web Apps 251
CHAPTER 6 Design a management, monitoring, and business continuity strategy 305

Index 381
Contents

Introduction xv

Microsoft certifications xv
Acknowledgments xvi
Free ebooks from Microsoft Press xvi
Microsoft Virtual Academy xvi
Errata, updates, & book support xvi
We want to hear from you xvii
Stay in touch xvii
Preparing for the exam xviii

Chapter 1  Design Microsoft Azure infrastructure and networking 1

Objective 1.1: Describe how Azure uses Global Foundation Services (GFS) datacenters 2

Azure’s global footprints 2
Designing cloud-scale datacenters 4
Designing for the cloud 8
Objective summary 11
Objective review 12

Objective 1.2: Design Azure virtual networks, networking services, DNS, DHCP, and IP addressing configuration 12

Creating a cloud-only virtual network 13
Understanding Access Control Lists and Network Security Groups 18
Objective summary 22
Objective review 22

What do you think of this book? We want to hear from you!
Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/
Chapter 2  Secure resources  

Objective 2.1: Secure resources by using managed identities  
- Understanding claims-based architecture  
- Understanding basic authentication and authorization workflow  
- Working with native clients and multitiered applications  
- Working with multitiered applications  
- Additional scenarios  
- Azure Active Directory  
- A sample scenario with ADAL and Visual Studio  
- Azure AD Graph API  
- Objective summary  
- Objective review  

Objective 2.2: Secure resources by using hybrid identities  
- Setting up directory synchronization with AD FS  
- Configuring Azure AD Application Proxy  
- Objective summary  
- Objective review  

Objective 2.3: Secure resources by using identity providers  
- Understanding Azure ACS  
- Using Azure ACS with AD FS  
- Using Azure ACS with social networks  
- Using identity providers with ASP.NET applications  
- Using external identity providers with Azure Mobile Services  
- Objective summary  
- Objective review  

Objective 2.4: Identify an appropriate data security solution  
- Understanding data protection technologies  
- Implementing effective access control policies  
- Using data reliability and disaster recovery services  
- Understanding Azure Rights Management Services  
- Managing security keys with Azure Key Vault  
- Objective summary  
- Objective review
Objective 2.5: Design a role-based access control strategy .......... 109

Understanding access control challenges faced by large enterprises 109
Implementing RBAC 110
Using RBAC for Azure resources 111
Empowering users with self-service 112
Using Azure AD Access Panel 115
Managing devices with Azure AD Device Registration Service 116
Improving security policies over time 117
Objective summary 120
Objective review 120

Answers. .................................................................................. 122
Objective 2.1: Thought experiment 122
Objective 2.1: Review 122
Objective 2.2: Thought experiment 123
Objective 2.2: Review 123
Objective 2.3: Thought experiment 124
Objective 2.3: Review 124
Objective 2.4: Thought experiment 125
Objective 2.4: Review 126
Objective 2.5: Thought experiment 127
Objective 2.5: Review 127

Chapter 3  Design an application storage and data access strategy  129

Objective 3.1: Design data storage ......................................... 129
Designing storage options for data 130
Designing security options for SQL Database or Storage 136
Identifying the appropriate VM type and size for the solution 137
Objective summary 139
Objective review 140

Objective 3.2: Design applications that use Mobile Services .......... 141
Azure Mobile Services 141
Consuming Mobile Services 143
Contents

Offline Sync 145
Implementing Mobile Services 147
Secure Mobile Services 148
Extending Mobile Services by using custom code 150
Objective summary 151
Objective review 151

Objective 3.3: Design applications that use notifications . . . . . . . . . . . . . . . . 153
Implementing push notification services in Mobile Services 153
Sending push notifications 155
Objective summary 157
Objective review 157

Objective 3.4: Design applications that use a web API . . . . . . . . . . . . . . . . . . . . . . . 158
Implementing a custom Web API 159
Scaling by using Azure App Service Web Apps 161
WebJobs 163
Securing a Web API 165
Objective summary 167
Objective review 168

Objective 3.5: Design a data access strategy for hybrid applications . . . . . . . . . 168
Connect to on-premises data by using Azure Service Bus Relay 169
Azure App Service BizTalk API Apps Hybrid Connections 170
Web Apps virtual private network capability 171
Identify constraints for connectivity with VPN 172
Identify options for domain-joining Azure Virtual Machines and Cloud Services 172
Objective summary 174
Objective review 174

Objective 3.6: Design a media solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 175
Azure Media Services overview 175
Key components of Media Services 176
Objective summary 179
Objective review 179
Chapter 4  Design an advanced application  189

Objective 4.1: Create compute-intensive applications  190
  Using Azure in a high-performance computing environment  190
  Using Azure Batch  193
  Understanding Azure Batch Apps  199
  Implementing the Competing Consumers pattern  200
  Objective summary  202
  Objective review  202

Objective 4.2: Create long-running applications  203
  Designing available applications  203
  Designing reliable applications  207
  Designing scalable applications  211
  Using Azure Autoscale  213
  Using Cloud Services  215
  Sample scenario: Cloud Services basics  217
  Objective summary  220
  Objective review  220

Objective 4.3: Select the appropriate storage option  221
  Understanding data access patterns  222
  Selecting a data storage solution  224
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Service Plan</td>
<td>273</td>
</tr>
<tr>
<td>Deployment slots</td>
<td>275</td>
</tr>
<tr>
<td>Resource groups</td>
<td>277</td>
</tr>
<tr>
<td>Publishing options</td>
<td>278</td>
</tr>
<tr>
<td>Objective summary</td>
<td>285</td>
</tr>
<tr>
<td>Objective review</td>
<td>285</td>
</tr>
<tr>
<td>Objective 5.3: Design websites for business continuity</td>
<td>286</td>
</tr>
<tr>
<td>Scale-up and scale-out with App Service Web Apps and Azure SQL Database</td>
<td>287</td>
</tr>
<tr>
<td>Configure data replication patterns</td>
<td>289</td>
</tr>
<tr>
<td>Update websites with minimal downtime</td>
<td>291</td>
</tr>
<tr>
<td>Backup and restore data</td>
<td>291</td>
</tr>
<tr>
<td>Design for disaster recovery</td>
<td>294</td>
</tr>
<tr>
<td>Deploy websites to multiple regions for high availability</td>
<td>294</td>
</tr>
<tr>
<td>Design data tier</td>
<td>296</td>
</tr>
<tr>
<td>Objective summary</td>
<td>298</td>
</tr>
<tr>
<td>Objective review</td>
<td>299</td>
</tr>
<tr>
<td>Answers</td>
<td>300</td>
</tr>
<tr>
<td>Objective 5.1: Thought experiment</td>
<td>300</td>
</tr>
<tr>
<td>Objective 5.1: Review</td>
<td>300</td>
</tr>
<tr>
<td>Objective 5.2: Thought experiment</td>
<td>301</td>
</tr>
<tr>
<td>Objective 5.2: Review</td>
<td>302</td>
</tr>
<tr>
<td>Objective 5.3: Thought experiment</td>
<td>303</td>
</tr>
<tr>
<td>Objective 5.3: Review</td>
<td>303</td>
</tr>
<tr>
<td>Chapter 6 Design a management, monitoring, and business continuity strategy</td>
<td>305</td>
</tr>
<tr>
<td>Objective 6.1: Evaluate hybrid and Azure-hosted architectures for Microsoft System Center deployment</td>
<td>306</td>
</tr>
<tr>
<td>Understanding System Center components supported in Azure</td>
<td>306</td>
</tr>
<tr>
<td>System Center deployment</td>
<td>309</td>
</tr>
<tr>
<td>Design considerations for managing Azure resources with System Center</td>
<td>310</td>
</tr>
</tbody>
</table>
Understanding which scenarios dictate a hybrid scenario 312
Objective summary 315
Objective review 315

Objective 6.2: Design a monitoring strategy 316
Identify the Microsoft products and services for monitoring Azure solutions 317
Understand the capabilities of System Center for monitoring an Azure solution 317
Understand built-in Azure capabilities 322
Identify third-party monitoring tools, including open source 323
Describe use-cases for Operations Manager, Global Service Monitor, and Application Insights 323
Describe the use cases for WSUS, Configuration Manager, and custom solutions 325
Describe the Azure architecture constructs and how they affect a patching strategy 327
Objective summary 329
Objective review 330

Objective 6.3: Design Azure business continuity/disaster recovery (BC/DR) capabilities 330
Understand the architectural capabilities of business continuity and disaster recovery 331
Describe Hyper-V Replica and Azure Site Recovery 336
Describe use-cases for Hyper-V Replica and Azure Site Recovery 338
Objective summary 340
Objective review 340

Objective 6.4: Design a disaster recovery strategy 341
Design and deploy Azure Backup and other Microsoft backup solutions for Azure 342
Understand use-cases when StorSimple and Data Protection Manager would be appropriate 351
Objective summary 352
Objective review 353
Introduction

Most books take a very low-level approach, teaching you how to use individual classes and accomplish fine-grained tasks. This book takes a high-level architectural view, building on your knowledge of lower-level Microsoft Azure systems management experience and content. Both the exam and the book are so high-level, in fact, that there are very few step-by-step instructions involved. There is some coding (Windows PowerShell and Azure PowerShell) but it is minimized to getting started with managing Azure with PowerShell and an introduction to how you can use Windows and Azure PowerShell to design, build, and deploy systems in the Azure cloud. The Exam Ref has a huge advantage over other study mechanisms: It demonstrates what is on the exam while also helping you to understand what it takes to design systems in real-world scenarios.

This book covers every exam objective, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions themselves and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the links you’ll find in the text to gain access to more information and take the time to research and study the topic. Great information is available on MSDN, TechNet, and in blogs and forums.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies, both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

MORE INFO  ALL MICROSOFT CERTIFICATIONS

For information about Microsoft certifications, including a full list of available certifications, go to http://www.microsoft.com/learning.
Acknowledgments

It takes many people to make a book, and even more to make a technical exam reference. Thanks to the content authors:

- Haishi Bai (http://haishibai.blogspot.com)
- Steve Maier (http://42base13.net)
- Dan Stolts (http://ITProGuru.com)

You can visit them and follow them on their blogs. Thanks to content providers and editors: Karen Szall, Devon Musgrave, Roberto Freato, and Bob Russell. Thanks to all those at Microsoft Press and Microsoft Learning for driving this certification, content, and resulting book. Most of all, thank you, for taking the time to learn about Azure cloud architecture through this exam reference guide.

Free ebooks from Microsoft Press

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

http://aka.ms/mspressfree

Check back often to see what is new!

Microsoft Virtual Academy

Build your knowledge of Microsoft technologies with free expert-led online training from Microsoft Virtual Academy (MVA). MVA offers a comprehensive library of videos, live events, and more to help you learn the latest technologies and prepare for certification exams. You’ll find what you need here:

http://www.microsoftvirtualacademy.com

Errata, updates, & book support

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

http://aka.ms/ER534/errata

If you discover an error that is not already listed, please submit it to us at the same page.
If you need additional support, email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to http://support.microsoft.com.

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

http://aka.ms/tellpress

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

Stay in touch
Let’s keep the conversation going! We’re on Twitter: http://twitter.com/MicrosoftPress.
Preparation for the exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Exam ref and another study guide for your "at home" preparation, and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you.

Note that this Exam Ref is based on publicly available information about the exam and the authors' experience. To safeguard the integrity of the exam, authors do not have access to the live exam questions.
CHAPTER 1

Design Microsoft Azure infrastructure and networking

What is the cloud? Among all the possible definitions, one captures the essence of the cloud in the simplest way: “The cloud is a huge pool of resources that supports a variety of services.”

The foundation of the cloud is a large pool of storage, compute, and networking resources. A key value proposition of the cloud is that you can acquire any amount of these resources at any time, from anywhere, without needing to worry about managing any underlying infrastructures. And when you are done with these resources, you can return them to the cloud just as easily to avoid the unnecessary cost to keep them around.

You can run services on top of these resources. Some of the services give you access to the infrastructure, such as virtual machines (VMs) and virtual networks—these services are called Infrastructure as a Service (IaaS). Some of the services provide support for building your own services in the cloud—these services are called Platform as a Service (PaaS). On top of IaaS and PaaS run Software as a Service (SaaS), which handle all kinds of workloads in the cloud.

After presenting a brief introduction of Microsoft Azure datacenters, this chapter focuses mostly on IaaS. It introduces tools and services for managing compute and network resources. In addition, it discusses design considerations and patterns to orchestrate these resources into complete solutions.

Objectives in this chapter:

- Objective 1.1: Describe how Azure uses Global Foundation Services (GFS) datacenters
- Objective 1.2: Design Azure virtual networks, networking services, DNS, DHCP, and IP addressing configuration
- Objective 1.3: Design Azure Compute
- Objective 1.4: Describe Azure virtual private network (VPN) and ExpressRoute architecture and design
- Objective 1.5: Describe Azure services
Objective 1.1: Describe how Azure uses Global Foundation Services (GFS) datacenters

To serve more than 1 billion customers across more than 140 countries and regions, Microsoft has built huge datacenters that have a combined total of more than 1 million servers. These datacenters are strategically placed at different geographic locations and are connected by high-performance fiber-optic networks. They provide continuous supports to more than 200 cloud services, such as Microsoft Bing, Office 365, OneDrive, Xbox Live, and Azure platform.

Managing enormous resource pools is not an easy task. Microsoft has invested tremendous resources to build reliable, secure, and sustainable datacenters. The team that manages and runs Azure infrastructure is called Microsoft Cloud Infrastructure and Operations (MCIO), formerly known as Global Foundation Service (GFS). This objective goes behind the scenes and reveals how these datacenters are designed, built, and maintained.

This section covers the following topics:
- Learning about Azure’s global footprints
- Understanding the design of cloud-scale data centers
- Designing for the cloud

EXAM TIP
You might find both MCIO and GFS are used in documentation, online materials, and white papers to refer to the team that operates Azure datacenters. As far as the exam is concerned, the two names are interchangeable. Also, sometimes Azure datacenters are referred to as Microsoft datacenters. The exam doesn’t distinguish between the two, either.

Azure’s global footprints
Azure is available in 140 countries and supports 10 languages and 19 currencies. Massive datacenters at 17 geographic regions provide scalable services to all Azure customers around the globe. For example, Azure Storage stores more than 30 trillion objects and serves on average in excess of 3 million requests per second.
MORE INFO  INCREASING NUMBERS

Back in 2012, Azure Storage maintained 4 trillion objects and served on average 270,000 requests per second (http://azure.microsoft.com/blog/2012/07/18/windows-azure-storage-4-trillion-objects-and-counting/). The number of objects grew to 20 trillion in early 2014 (http://azure.microsoft.com/en-us/documentation/videos/microsoft-storage-what-new-best-practices-and-patterns/), and then it reached 30 trillion later that same year. But these exact numbers don’t matter all that much; what’s important is to realize the rapid growth of Azure.

Regions and datacenters

Azure operates in 17 regions. Each region contains one or more datacenters. Table 1-1 lists current Azure regions and their corresponding geographic locations.

<table>
<thead>
<tr>
<th>Azure region</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central US</td>
<td>Iowa</td>
</tr>
<tr>
<td>East US</td>
<td>Virginia</td>
</tr>
<tr>
<td>East US 2</td>
<td>Virginia</td>
</tr>
<tr>
<td>US Gov Iowa</td>
<td>Iowa</td>
</tr>
<tr>
<td>US Gov Virginia</td>
<td>Virginia</td>
</tr>
<tr>
<td>North Central US</td>
<td>Illinois</td>
</tr>
<tr>
<td>South Central US</td>
<td>Texas</td>
</tr>
<tr>
<td>West US</td>
<td>California</td>
</tr>
<tr>
<td>North Europe</td>
<td>Ireland</td>
</tr>
<tr>
<td>West Europe</td>
<td>Netherlands</td>
</tr>
<tr>
<td>East Asia</td>
<td>Hong Kong SAR</td>
</tr>
<tr>
<td>Southeast Asia</td>
<td>Singapore</td>
</tr>
<tr>
<td>Japan East</td>
<td>Saitama Prefecture</td>
</tr>
<tr>
<td>Japan West</td>
<td>Osaka Prefecture</td>
</tr>
<tr>
<td>Brazil South</td>
<td>Sao Paulo State</td>
</tr>
<tr>
<td>Australia East</td>
<td>New South Wales</td>
</tr>
<tr>
<td>Australia Southeast</td>
<td>Victoria</td>
</tr>
</tbody>
</table>
Be aware that in some texts the terms “regions” and “locations” are often used interchangeably. A datacenter is also sometimes referred as a facility. Azure doesn’t have a formal concept of “zones,” although a zone roughly maps to a datacenter or a facility in some contexts. For example, Azure Storage provides Zone-Redundant Storage (ZRS), which maintains three copies of your data across two to three facilities within a single region or across two regions.

Another concept regarding compute resource placements is the Affinity Group. Affinity Group is a way to group your cloud services by proximity to each other in an Azure datacenter to minimize communication latency. When you put your services in the same Affinity Group, Azure knows that they should be deployed on hardware that is close to one another to reduce network latency.

**MORE INFO STAMPS**

In some online literatures, you might also see references to stamps. A stamp loosely refers to a group of server racks. It’s not an official concept and is never stipulated as a management or deployment boundary.

**Regional differences**

Not all Azure regions provide the same set of services. As a new service is being rolled out, it might at first become available only at a small set of regions and then become available across all regions. Some regions have additional constraints. For example, the Australia regions are available only to customers with billing addresses in Australia and New Zealand. For a complete region/service cross-reference table, go to http://azure.microsoft.com/en-us/regions/#services.

Azure is available in China. However, you might have noticed that China is not listed as one of the regions in Table 1-1. This is because Azure in China is independently operated by 21Vianet, one of the largest Internet Service Providers (ISPs) in China. Your Azure subscriptions provisioned for the China region cannot be used for other regions. The reverse is also true: your subscriptions outside the China region cannot be used for the China region.

Azure’s multilanguage support is not tied to specific regions. You can choose your Azure Management Portal language as a user preference. For example, it’s perfectly fine to use a user interface (UI) localized in Japanese to manage resources around the globe. However, many Azure objects don’t allow non-English characters in their names or identifiers.

**Designing cloud-scale datacenters**

A single Azure datacenter can be as big as three large cruise ships placed end to end and host tens of thousands of servers. This level of unprecedented scale brings additional challenges in datacenter design and management. A radically different strategy is needed to design and operate cloud-scale datacenters.
Embracing errors

Cloud-scale datacenters use commodity servers to reduce cost. The availability of these servers is often not as high as the more expensive ones you see in traditional datacenters. And when you pack hundreds of thousands of servers and switches into the same facility, hardware failures become the norm of day-to-day operation. It’s unimaginable to remedy these failures individually. A different approach is needed.

Traditionally, datacenter designs focus on increasing Mean Time between Failures (MTBF). With a few servers available to host certain workloads, each of the servers is required to be highly reliable so that a healthy server can remain online for an extended period of time when a failing server is being repaired or replaced. With commodity servers, such long MTBF can’t be guaranteed. However, cloud-scale datacenters do have an advantage: they have lots of servers. When one server is failing, its workloads can be directed to another healthy server for recovery. This workload migration mechanism makes it possible for customer services to recover from hardware failures quickly. In other words, cloud-scale datacenters focus more on Mean Time to Recover (MTTR) instead of MTBF, because, in the end, what customers care about is the availability of their services, not the availability of underlying hardware.

Due to the sheer number of servers, such workload migrations can’t happen manually in cloud-scale datacenters. To bring MTTR to its minimum requirement, automation is the key. Errors must be detected and handled automatically so that they can be fixed with minimum delays.

Human factors

When it comes to following rules and avoiding mistakes, humans are much less reliable than machines. Unfortunately, humans have the ultimate controlling power over all machines (or so it seems in the present day). Looking back a bit, some of the massive outages in cloud-scale datacenters were caused by humans. As the saying goes, to err is human, and such mistakes will happen, regardless of what countermeasures have been put in place. However, there are some key strategies that can help cloud-scale datacenters to reduce such risks.

Abundant training, rigorous policy reinforcement, continuous monitoring, and auditing form the foundation of an error-resilient team. However, using privileged accounts still has its inherent risks. Azure adopts polices such as just-in-time administrator accesses and just-enough administrator accesses. Microsoft staff doesn’t have access to customer data by default. When Microsoft personnel need access to Azure resources for diagnosing specific customer problems, they are granted access to the related resources for no more than a predetermined window. All activities are carefully monitored and logged. At the same time, Azure also encourages customers managing their accesses to resources to follow best practices by providing tools, services, and guidance such as Azure Active Directory (Azure AD) multifactor authentication, built-in Role-Based Access Control (RBAC) with Azure Resource Groups, and Azure Rights Management.
Automation is undoubtedly one of the most effective means to reduce human errors. Azure provides several automation options, including Azure Management API, Azure PowerShell, and Azure Cross-Platform Command-Line Interface (xplat-cli). In addition, Azure also provides managed automation services such as Azure Automation, which is covered in Chapter 6. In terms of automating resource state management at scale, you can use first-party solutions such as Custom Script Extension and Windows PowerShell Desired State Configuration (DSC), or use integrated third-party solutions such as Puppet and Chef.

**Trust-worthy computing**

Although the adoption of the cloud has been accelerating, many organizations still have doubts when it comes to handing their valuable business data and mission-critical workloads to a third party. Cloud platforms such as Azure need to work with the highest standards and greatest transparency to build their credibility as trust-worthy business partners. This is a challenge not unique to Azure, but to the entire cloud industry.

It is the policy of Microsoft that security, privacy, and compliance are a shared responsibility between Azure and Azure’s customers. Azure takes over some of the burden for implementing operational processes and technical safeguards, including (but not limited to) the following:

- **Physical security and continuous surveillance.**
  Azure datacenters are protected by physical barriers and fencing, with integrated alarms, cameras and access controls. The facilities are constantly monitored from the operations center.

- **Protection against virus, malware, and DDoS attacks.**
  Azure scans all software components for malware and viruses during internal builds and deployments. Azure also enables real-time protection, on-demand scanning and monitoring for Cloud Services and VMs. To prevent attacks such as DDoS, Azure performs big data analysis of logs to detect and respond to intrusion risks and possible attacks.

- **Activity monitoring, tracing and analysis, and abnormality detection.**
  Security events are continuously monitored and analyzed. Timely alerts are generated so that hardware and software problems can be discovered and mitigated early.

- **System patching, such as applying security patches.**
  When patch releases are required, they are analyzed and applied to the Azure environment based on the severity. Patches are also automatically applied to customer guest VMs unless the customer has chosen manual upgrades, in which case the customer is responsible for patching.

- **Customer data isolation and protection.**
  Azure customers are logically isolated from one another. An Azure customer has no means to access another customer’s data, either intentionally or unintentionally. We cover data protection in more detail in Chapter 2.
On the other hand, Azure provides tools and services to help customers to realize their own security and compliance goals. A good example is data encryption for Azure Storage. Azure offers a wide range of encryption options to protect data at rest. Azure also provides a Key Vault service to manage security keys. However, it’s up to the customers to make appropriate choices based on their security and performance requirements. The customers must decide which technologies to use and how to balance between security and performance overheads. Furthermore, customers need to utilize security communication channels such as SSL and TLS to protect their data during transition.

To help customers to achieve compliance goals, Microsoft has developed an extensible compliance framework by which Azure can adapt to regulatory changes. Azure has been independently verified by a diverse range of compliance programs, such as ISO 27001/27002, FISMA, FedRAMP, HIPPA, and EU Model Clauses.

**Sustainable reliability**

Each of the Azure datacenters hosts a large number of services. Many of these are mission-critical services that customers rely on to keep their businesses running. There’s a lot at stake for both Microsoft and its customers. So, the very first mission of Azure datacenter design is to ensure infrastructure availability. For critical infrastructural components such as power supplies, Azure builds multiple levels of redundancies. Azure datacenters are equipped with Uninterruptible Power Supply (UPS) devices, massive battery arrays, and generators with on-site fuel reserves to ensure uninterrupted power supply even during disastrous events.

These extreme measures incur significant cost. Azure datacenters must be carefully designed so that such additional layers of protections can be provided while the total cost of ownership is still well controlled. Microsoft takes a holistic approach to optimize its datacenters. Instead of focusing on optimizing a single component, the entire ecosystem is considered as a whole so that the Total Cost of Ownership (TCO) remains low without compromising efficiency.

As a matter of fact, Microsoft runs some of the most efficient cloud-scale datacenters in the world with Power Usage Effectiveness (PUE) measures as low as 1.125. PUE is the ratio between total facility power usage and IT equipment’s power usage. A lower PUE means less power is consumed to support day-to-day facility operations such as providing office lighting and running elevators. Because such additional power consumption is unavoidable, a PUE of 1.125 is very hard to achieve. For comparison, the industry norm is about 1.8.
Last but not least, Azure datacenters are environment-friendly. Microsoft is committed to reducing the environmental footprint of its datacenters. To make these datacenters sustainable, Microsoft has implemented a comprehensive strategy that involves every aspect of datacenter design and operation, such as constructing datacenters using recycled materials, utilizing renewable power sources, and pioneering in efficient open-air cooling.

Since its first datacenter was constructed in 1989, Microsoft has never stopped innovating how datacenters are designed and operated. Four generations later, Azure datacenters are looking forward to the next new generation of datacenters—and they’re just on the horizon—which will be even more efficient and sustainable. The benefits of these innovations are passed to Azure’s customers and eventually billions of end users around the world.

**Designing for the cloud**

The unique characteristics of cloud-scale datacenters bring both challenges and opportunities to designing your applications. On one hand, you need to ensure that your application architecture is adapted for these characteristics so that your application can function correctly. On the other hand, you want to take advantage of Quality of Service (QoS) opportunities that the cloud offers, allowing your applications to thrive.

This section focuses on the first aspect, which is to ensure that your applications function correctly in cloud-scale datacenters. Chapter 4 discusses how to improve QoS in the cloud.

**Datacenter maintenance**

Azure performs two types of maintenances: planned and unplanned. Planned maintenance occurs periodically on a scheduled basis; unplanned maintenance is carried out in response to unexpected events such as hardware failures.

**PLANNED MAINTENANCE**

Azure periodically performs maintenance on the hosting infrastructure. Many of these maintenances occur at the hosting operation system level and the platform software level without any impact to hosted VMs or cloud services. However, some of these updates will require your VMs to be shut down or rebooted.

You can configure VMs on Azure in two ways: multi-instance and single-instance. Multi-instance VMs are joined to a same logical group called an *Availability Set*. When Azure updates VMs, it guarantees that not all machines in the same Availability Set will be shut down at the same time. To ensure your application availability, you should deploy your application on an Availability Set with at least two VMs. Only multi-instance VMs qualify for the Service Level Agreement (SLA) provided by Azure.

**MORE INFO UPDATE DOMAIN AND FAULT DOMAIN**

Two concepts related to Availability Set are *Update Domain* and *Fault Domain*. Chapter 4 introduces these two concepts in more detail within the context of service availability and reliability.
Single-instance VMs are stand-alone VMs. During datacenter updates, these VMs are brought down in parallel, upgraded, and brought back online in no particular order. If your application is deployed on a single-instance VM, the application will become unavailable during this maintenance window. To help preclude any problems, Microsoft sends email notices to single-instance customers, indicating the exact date and time on which the maintenance is scheduled, as shown in Figure 1-1. Thus, if your Availability Set contains only a single VM, the availability of your application will be affected because there will be no running instances when the only machine is shut down.

**FIGURE 1-1** A sample maintenance notification email

**NOTE** AVOID HAVING A SINGLE VM IN AN AVAILABILITY SET

A single VM in an Availability Set doesn’t qualify for SLA. Azure requires at least two VMs to be deployed in an Availability Set in order to qualify for SLA.
UNPLANNED MAINTENANCE
Unplanned maintenances are triggered by unexpected physical infrastructure problems such as network failures, rack-level failures and other hardware failures. When such a failure is detected, Azure automatically moves your VMs to a healthy host. When multiple VMs are deployed in the same Availability Set, they are allocated to two Fault Domains (you can read more on this in Chapter 4). At the hardware level, Fault Domains don’t share a common power source or network switch, so the probability of two Fault Domains failing at the same time is low.

Azure’s autorecovery mechanism significantly reduces MTTR. In traditional datacenters, recovering or replacing a server often needs a complex workflow that can easily take days or even weeks. By comparison, Azure can recover a VM in minutes. Regardless of how short the window is, the VM is still restarted. Your application needs to be able to restart itself when this happens. Otherwise, although the VM is recovered, your application is still unavailable.

Azure Cloud Service has a built-in mechanism to monitor and recover your application process. For applications deployed on VMs, you can define endpoints with load-balanced sets. A load-balanced set supports custom health probes, which you can use to detect if your application is in running state. Load-balanced sets are discussed further in Objective 1.3.

Datacenter outages
No cloud platform is immune to some large-scale outages caused by natural disasters and occasionally human errors. Microsoft has adopted a very transparent policy that shares very thorough Root Cause Analysis (RCA) reports with customers when such outages happen. These reports disclose the exact cause of the outage, no matter if it is because of code defects, architecture flaws, or process violations. Microsoft works very hard to ensure that the mistake is not repeated in the future.

Cross-region redundancy is an effective way to deal with region-wide outages. Later in this book, you’ll learn technologies such as Azure Traffic Manager and Service Bus paired namespaces that help you to deploy cross-region solutions.

Service throttling
The cloud is a multitenant environment occupied by many customers. To ensure fair resource consumption, Azure throttles service calls according to subscription limits. When throttling occurs, you experience degraded services and failures in service calls.

Different Azure services throttle service calls based on different criteria, such as the amount of stored data, the number of transactions, and system throughputs. When you subscribe to an Azure service, you should understand how the service throttles your calls and ensure that your application won’t exceed those limits.

Most Azure services offer you the option to gain additional capacities by creating multiple service entities. If you’ve decided that a single service entity won’t satisfy your application’s needs, you should plan ahead to build multi-entity support into your architecture so that your application can be scaled out as needed.
Another effective way to offset some of the throttling limits is to use caches such as application-level caching and Content Delivery Networks (CDNs). Caches help you not only to reduce the amount of service calls, but also to improve your application performance by serving data directly from cache.

**Service security**

With the exception of a few read-only operations, Azure requires proper authentication information to be present before it grants a service request. Azure services supports three different authentication strategies: using a secret key, using a Shared Access Signature (SAS), and using federated authentication via Azure AD.

When a secret key is used, you need to ensure that the key itself is securely stored. You can roll out a protection strategy yourself, such as using encryptions. Later in this chapter, you’ll see how Azure Key Vault provides an efficient, reliable solution to this common problem.

SAS is a proven way to provide detailed level of access control over entities. With SAS, you can grant access to specific data with explicit rights during given time windows. The access is automatically revoked as soon as the window is closed.

Azure AD is discussed in depth in Chapter 2.

---

**Thought experiment**

**Explaining the benefits of cloud**

In this thought experiment, apply what you’ve learned about this objective. You can find answers to these questions in the “Answers” section at the end of this chapter.

Although cloud adoption has been accelerating over the past few years, many enterprise decision makers remain very cautious when deciding on cloud strategies. In particular, they are concerned about data security and service reliability. They have doubts when it comes to handling valuable business data to a third-party. And their doubts are reinforced by occasional news outbursts on cloud datacenter outages and breaches. As a technical lead, you need to come up with a strategy to convince these decision makers to adopt a cloud strategy.

With this in mind, answer the following questions:

1. How would you explain the benefits of the cloud in terms of data security?
2. How would you explain the benefits of the cloud in terms of reliability?

---

**Objective summary**

- Azure serves more than 1 billion customers out of 17 global locations. Azure runs more than 200 online services in more than 140 countries.
A key strategy to improve service availability in the cloud is to reduce MTTR. Workload is reallocated to healthy servers so that the service can be recovered quickly.

Automation, just-in-time access, and just-enough access are all effective ways to reduce possible human errors.

Azure datacenters take over some of the responsibilities of infrastructure management by providing trustworthy and sustainable infrastructures.

Your application needs to be designed to cope with service interruptions and throttling. In addition, your application needs to adopt appropriate security policies to ensure that your service is only accessed by authenticated and authorized users.

Objective review

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of this chapter.

1. Which are the effective ways to reduce human errors?
   A. Sufficient training
   B. Automation
   C. Just-in-time access
   D. Reinforced operation policy

2. Azure has been independently verified by which of the following compliance programs?
   A. ISO 27001/27002
   B. FedRAMP
   C. HIPPA
   D. EU Model Clauses

3. Which of the following VM configurations qualifies for availability SLA?
   A. Single-instance VM
   B. Multi-instance VMs on an Availability Set
   C. Single-instance VM on an Availability Set
   D. Two single-instance VMs
Objective 1.2: Design Azure virtual networks, networking services, DNS, DHCP, and IP addressing configuration

Today, just about any computer you see is connected to some network. Computers on Azure are no exception. When you provision a new VM on Azure, you never gain physical access to the hosting machine. Instead, you need to operate the machine through remote connections such as remote desktop or Secure Shell (SSH). This is made possible by the networking infrastructure provided by Azure.

This objective introduces Azure Virtual Networks, with which you can create virtualized private networks on Azure. VMs deployed on a virtual network can communicate with one another just as if they were on an on-premises local area network (LAN).

Furthermore, you can connect your virtual networks with your on-premises networks, or with other virtual networks, through cross-network connections. You’ll learn about hybrid networks in objective 1.4.

This section covers the following topics:
- Creating a cloud-only virtual network
- Understanding ACLs and Network Security Groups

Creating a cloud-only virtual network

It’s fairly easy to create a new virtual network on Azure. This section walks you through the steps to set up a new virtual network with two subnets on Azure. Then, you will review some of the differences between a virtual network and an on-premises network that you should be aware of when you design your network infrastructures in the cloud.

NOTE  REVIEW OF BASIC NETWORKING CONCEPTS

This objective doesn’t require readers to have deep networking knowledge. Instead, it assumes most readers don’t routinely maintain networks and might need refreshers of basic networking concepts. These concepts are explained as side notes throughout this chapter. Feel free to skip these notes if you are already familiar with the concepts.

Creating a virtual network by using the Azure management portal

There are several different ways you can create a new virtual network on Azure, including using the Azure management portal, Azure PowerShell, and xplat-cli. This section walks you through how to use the management portal to create a new virtual network. Scripting options are discussed later in this chapter.
2. Select New, Network Services, Virtual Network, and then Custom Create, as shown in Figure 1-2.

![Figure 1-2 Creating a new virtual network](image)

The Create A Virtual Network Wizard opens.

3. On the Virtual Network Details page, in the Name box, type a name for the virtual network. In the Location box, select a location where you want the network to reside. If you have multiple Azure subscriptions, you also need to pick which Azure subscription to use. Then, click the right-arrow button to continue, as illustrated in Figure 1-3.

![Figure 1-3 Specifying the virtual network name and location](image)

**NOTE ABOUT AFFINITY GROUPS FOR VIRTUAL NETWORKS**

Previously, when you created a virtual network, you needed to associate the network with an Affinity Group. This is no longer a requirement. Now, virtual networks are associated directly with a region (location). Such virtual networks are called regional virtual network in some texts. The previous requirement of having Affinity Groups was because Azure networks were designed in layers. Communication among hardware within the same “branch” was much faster than communication across branches. A new
flat network design makes it possible for VMs across the entire region to communicate effectively, eliminating the need to put a virtual network in an Affinity Group.

4. On the DNS Servers And VPN Connectivity page, click Next to continue. (You’ll come back to these options later in this chapter.)

The Virtual Network Address Spaces page opens, as shown in Figure 1-4.

When you manage a larger virtual network, you might want to create multiple subnets to improve performance. To describe this briefly, a network is like a web of roads. When you have more computers sending and receiving packets on the same network, packets can collide and must be resent again. Using subnets, you can control and limit traffic in different areas. It’s similar to using local roads for a short commute, and using shared highways to travel longer distances.

In many cases, subnets are created not only for performance, but also for manageability. You can create subnets in alignment with business groups. For example, you can create one subnet for the sales department, and another subnet for engineering. You can also create subnets based on server roles. For example, you can have a subnet for web servers and another subnet for file servers.

**NOTE ABOUT CIDR NOTATION**

Classless Inter-Domain Routing (CIDR) notation is a shorthand representation of a subnet mask. It uses the number of bits to represent a subnet mask. For example, a subnet mask of 255.0.0.0 uses 8 bits, hence it’s written as /8. And a subnet mask of 255.255.0.0 uses 16 bits, which is written as /16 in CIDR notation. With CIDR, 10.0.0.0/8 in Figure 1-4...
represents a network ID of 10.0.0.0 and a subnet mask of 255.0.0.0, which corresponds to the address range 10.0.0.0 to 10.255.255.255.

5. Click the Add Subnet button to create a new subnet. The updated address space is illustrated in Figure 1-5. In the lower-right corner, click the check button to complete the set up.

<table>
<thead>
<tr>
<th>ADDRESS SPACE</th>
<th>STARTING IP</th>
<th>CIDR (ADDRESS COUNT)</th>
<th>USABLE ADDRESS RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/8</td>
<td>10.0.0.0</td>
<td>/8 (16777...)</td>
<td>10.0.0.0 - 10.255.255</td>
</tr>
<tr>
<td>SUBNETS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subnet-1</td>
<td>10.0.0.0</td>
<td>/11 (2097...)</td>
<td>10.0.0.0 - 10.31.255</td>
</tr>
<tr>
<td>Subnet-2</td>
<td>10.32.0.0</td>
<td>/11 (2097...)</td>
<td>10.32.0.0 - 10.63.255</td>
</tr>
</tbody>
</table>

**FIGURE 1-5** A virtual network with two subnets

Now, your network has two subnets, each has 2,097,152 (2²¹) addresses.

**NOTE ABOUT SUBNET BITS AND THE NUMBER OF SUBNETS**

When you create a subnet, you are borrowing a number of bits from the host ID and adding them to the network ID. In previous example, we are borrowing 3 bits, which means you can create up to 8 (2³) subnets. Because the bits borrowed are high bits, they correspond to 0, 32, 64, 96, 128, 160, 192, and 224. This is why the first IP address on the second subnet is 10.32.0.0.

**IP Addresses**

Each VM has at least two associated IP addresses: a public-facing virtual IP (VIP) address, and an internal dynamic IP (DIP) address.

A VIP comes from a pool of IP addresses managed by Microsoft. It is not assigned directly to the VM. Instead, it’s assigned to the Cloud Service that contains the VM. You can reserve VIPs so that you can assign static public IPs to your VMs. At this point, each Azure subscription is allowed to reserve up to 20 VIPs.

**NOTE ABOUT VMS AND CLOUD SERVICES**

Each VM you create belongs to a cloud service. Cloud Services is introduced in Chapter 4. For now, you can understand a cloud service as a management and security boundary for VMs. VMs residing in the same cloud service have a logical private scope, within which they can communicate with one another directly using host names.

To reserve a VIP, use the following Azure PowerShell command:
New-AzureReservedIP -ReservedIPName "MyReservedIP" -Label "MyLabel" -Location "West US"

After you have the static VIP allocated, you can use it as part of the VM configuration when you create a new VM. VMs are discussed in the next objective.

The DIP address is a dynamic IP address associated with your VM. A DIP is assigned by DHCP with a near-infinite lease. So, it remains stable as long as you don’t stop or deallocate the machine. However, it’s not a static IP address. If your VM resides in a virtual network, you can assign a static IP address to it. For example, when you set up a domain controller or a Domain Name System (DNS) server on your virtual network, you’ll need to assign static IPs to these machines because both services require static IP addresses.

With Azure, you can create multiple virtual network interfaces (NICs) on your VM residing on a virtual network. In this case, your VM has multiple associated DIPs, one for each NIC.

In addition to VIP and DIP, there’s another type of IP address, which is called Instance-Level Public IP (PIP) Address. As stated previously, a VIP is not assigned to a VM, but to the Cloud Service containing the VM. A PIP, on the other hand, is directly assigned to a VM. PIP is appropriate for workloads that need a large number of ports to be opened, such as passive FTP.

**Name resolution and DNS servers**

VMs on the same network can address one another by DIP addresses. If you want to refer to VMs by hostnames or fully qualified domain name (FQDN) directly, you need name resolutions. Azure provides a built-in hostname resolution for VMs and role instances within the same cloud service. However, for VMs across multiple cloud services, you’ll need to set up your own DNS server.

**HOST NAMES AND FQDNS**

As is discussed in Objective 1.3, when you create a new VM, the host name is specified by you. And when you define a cloud service role (you can read more about Cloud Services in Chapter 4), you can define the VM host name by using the `vmName` property in the service configuration file. In this case, Azure will append an instance number to the name to distinguish different role instances. For example, if `vmName` is `MyRole`, the actual host names of role instances will be `MyRole01`, `MyRole02`, and so on.

When you create a VM (or a cloud service), a DNS name is assigned to the machine with the format `[machine name].cloudapp.net`, where `[machine name]` is the name you specify. You can use this FQDN to address your machine directly over Internet. When the VM is provisioned, a public-facing VIP is associated with the machine, and then the DNS name is associated with this VIP.

You can also use CNAME or A records to associate a custom domain name with your VM. When you use A records, however, you need to note that the VIP of your VM might change. When you deallocate a VM, the associated VIP is released. And when the VM is restarted later, a new VIP will be picked and assigned. If you want to ensure that your VM has a static public IP address, you’ll need to configure a static IP address for it as described earlier.
Last but not least, for simple name resolutions, you can also use hosts files (%System32%\Drivers\etc\hosts for Windows; /etc/hosts for Linux) and cross-enter IP-to-host mappings to all the VMs in the same virtual network.

DNS SERVERS
You can set up DNS servers on your virtual network to provide a name resolution service to the machines on the same network. Objective 1.3 presents a couple of examples.

Understanding Access Control Lists and Network Security Groups
You can use both network Access Control Lists (ACLs) and Network Security Groups (NSGs) to control traffic to your VMs. In either case, the traffic is filtered before it reaches your VM so that your machine doesn’t need to spend extra cycles on packet filtering.

Before you continue learning about ACLs and NSGs, you need to first understand how VM endpoints work.

VM endpoints
When you provision a VM on Azure by using the management portal, by default the device is accessible through Remote Desktop and Windows PowerShell Remoting for Windows-based VMs, and through SSH for Linux-based VMs. This is because Azure automatically defines the corresponding endpoints.

Each endpoint maps a public port to a private port. The private port is used by the VM to listen for incoming traffic. For Example, your device might have an Internet Information Services (IIS) server running on it, listening to the private port 80. The public port is not used by the VM itself, but by another entity called Azure Load Balancer.

As mentioned earlier, a VM has a VIP address as well as a DIP address. However, the VIP address is actually not directly associated with the VM. Instead, the VIP address is associated with Load Balancer. It’s Load Balancer that listens to the traffic to the VIP address and the public port, and then forwards the traffic to the VM listening to the DIP address and the private port. Figure 1-6 shows how this traffic forwarding works. At the top, the traffic reaches the endpoint at VIP:[public port]. Then, Load Balancer forwards the traffic to DIP:[private port]. In this example, an endpoint is defined to map a public port 8080 to a private port 80. The IIS server on a VM named myvm is listening to local address 10.0.0.1:80. An end user accesses the website by the public address myvm.cloudapp.net:8080. Note that the “myvm” in the FQDN “myvm.cloudap.net” is the name of the Cloud Service in which the VM resides. It’s not necessarily the same as the VM name (you can have multiple VMs in the same Cloud Service).
Endpoints can be stand-alone or load-balanced. When a load-balanced endpoint is defined, Load Balancer distributes traffic evenly among the VMs within the same load-balanced set. Figure 1-7 shows how it works.

Endpoints are for public accesses. When you provision a VM on a virtual network, it can communicate with other VMs on the same network just as if they were on a physical local network. There are no endpoints needed for private communications.
Network ACLs

ACL provides the ability to selectively permit or deny traffic to a VM endpoint. An ACL comprises an ordered list of rules that either permit or deny traffic to the endpoint. Packets are filtered on the hosting server before they can reach your VM. When a new endpoint is created, by default all traffic from all IP addresses are allowed. Then, you can define ACLs to constrain accesses to certain ranges of IP addresses by defining blocked lists and safe lists, each of which is defined here:

- **Blocked list**  You can block ranges of IP addresses by creating deny rules. Table 1-2 shows an example of ACL that blocks accesses from a specific subnet:

  **TABLE 1-2** A sample blocked list

<table>
<thead>
<tr>
<th>Rule #</th>
<th>Remote subnet</th>
<th>Endpoint</th>
<th>Permit/deny</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10.32.0.0/11</td>
<td>80</td>
<td>Deny</td>
</tr>
</tbody>
</table>

- **Safe list**  You can also create a safe list that allows only specific IP addresses to access an endpoint. First, you’ll define a rule that denies all traffic to the endpoint. Then, you add additional rules to allow accesses from specific IP addresses (ACL uses lowest takes precedence rule order). Table 1-3 shows a sample safe list:

  **TABLE 1-3** A sample safe list

<table>
<thead>
<tr>
<th>Rule #</th>
<th>Remote subnet</th>
<th>Endpoint</th>
<th>Permit/deny</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.0.0.0/0</td>
<td>80</td>
<td>Deny</td>
</tr>
<tr>
<td>200</td>
<td>10.0.0.0/11</td>
<td>80</td>
<td>Permit</td>
</tr>
</tbody>
</table>

You can apply ACLs to load-balanced endpoints, as well. When you apply an ACL to a load-balanced endpoint, it’s applied to all VMs in the same load-balanced set. You can specify up to 50 ACL rules per VM endpoint.

NSGs

For VMs deployed on a virtual network, NSGs provide more detailed access controls. An NSG is a top-level object of your Azure subscription that you can apply to a VM or a subnet to control traffic to the VM or the subnet. You can also associate different NSGs to a subnet and the VMs contained in the virtual network to establish two layers of protections.

Similar to an ACL, an NSG is made up by a number of prioritized rules. Each NSG comes with a number of default rules that you can’t remove. However, as these rules have lower priorities, you can override them by additional rules. There are two types of rules: inbound rules and outbound rules. Each rule defines whether the traffic should be denied or allowed to flow from a source IP range and port to a destination IP range and port. You can also specify protocols in NSG rules. The supported protocols are TCP and UDP, or * for both.

In NSG rules, IP ranges are represented by named tags. There are three default tags:
- **VIRTUAL_NETWORK** This tag specifies all network address space on your virtual network. It also includes connected on-premises address spaces and vNet-to-vNet address spaces (you’ll learn about on-premises connections and vNet-to-vNet connections in Objective 1.4).

- **AZURE_LOADBALANCER** This tag denotes Azure Load Balancer. Load Balancer sends health probe signals to VMs in a load-balanced set. This tag is used to identify the IP address from which the health probes originate.

- **INTERNET** This tag specifies all IP address that are outside the virtual network.

  With an NSG, inbound traffic is denied by the default rules, with the exception of allowing health probes from Load Balancer. Table 1-4 lists the default inbound rules of an NSG. The first rule allows all internal traffic within the same virtual network; the second rule allows health probes from Load Balancer; and the third rule denies all other accesses.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>Protocol</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>65000</td>
<td>VIRTUAL_NETWORK</td>
<td>*</td>
<td>VIRTUAL_NETWORK</td>
<td>*</td>
<td>*</td>
<td>Allow</td>
</tr>
<tr>
<td>65001</td>
<td>AZURE_LOADBALANCER</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Allow</td>
</tr>
<tr>
<td>65000</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Table 1-5 lists the default outbound rules of a NSG. The first rule allow outbound traffic to the virtual network. The second rule allows outbound traffic to Internet. And the third rule denies all other outbound traffic.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination IP</th>
<th>Destination Port</th>
<th>Protocol</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>65000</td>
<td>VIRTUAL_NETWORK</td>
<td>*</td>
<td>VIRTUAL_NETWORK</td>
<td>*</td>
<td>*</td>
<td>Allow</td>
</tr>
<tr>
<td>65001</td>
<td>*</td>
<td>*</td>
<td>INTERNET</td>
<td>*</td>
<td>*</td>
<td>Allow</td>
</tr>
<tr>
<td>65000</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Deny</td>
</tr>
</tbody>
</table>

NSGs are different from ACLs in a couple of aspects:

- ACLs are applied to traffic to a specific VM endpoint, whereas NSGs are applied to all traffic that is inbound and outbound on the VM.
- ACLs are associated to a VM endpoint, whereas NSGs are associated to a VM, or a subnet within a virtual network.

**NOTE INCOMPATIBILITY BETWEEN ACL AND NSG**

You cannot use both ACL and NSG on the same VM instance. You must first remove all endpoint ACLs before you can associate an NSG.
Thought experiment
Implementing perimeter networks in Azure Virtual Network

In this thought experiment, apply what you’ve learned about this objective. You can find answers to these questions in the “Answers” section at the end of this chapter.

Using isolated security zones is an effective way for enterprises to reduce many types of risks on their networks. For example, many enterprises use a perimeter network to isolate their Internet-facing resources from other parts of their internal network. You can implement the same level of protection in Azure Virtual Network, as well. In this case, you have a number of VMs that will be exposed to the Internet. And you have a number of application servers and database servers on the same virtual network.

With this in mind, answer the following questions:

1. What technologies would you use to implement a perimeter network in Virtual Network?
2. How would you design your network topology?

Objective summary

- You can create private virtual networks in Azure. VMs deployed on the same virtual network can communicate with one another as if they were on the same local network.
- Each machine has a public VIP address and one or multiple PIP addresses, one per NIC.
- You can associate both static virtual IP addresses and static private IP addresses to VMs on a virtual network.
- ACLs are associated to VM endpoints to control traffic to VMs.
- NSGs are associated to VMs or subnets to provide greater traffic control to VMs or virtual networks.
- Both ACLs and NSGs define prioritized rules to control network traffic, but they cannot be used in conjunction.

Objective review

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of this chapter.
1. A VM can have multiple associated IP addresses. Which of the following are possible IP addresses associated with a VM?
   A. Public virtual IP  
   B. Dynamic private IP  
   C. Static public IP  
   D. Static private IP  

2. NSGs define a number of default tags. Which of the following tags are default tags?
   A. VIRTUAL_NETWORK  
   B. AZURE_LOADBALANCER  
   C. INTERNET  
   D. VIRTUAL_MACHINE  

3. Which of the following are NSG rule fields?
   A. Source IP and source port  
   B. Target IP and target port  
   C. Protocol  
   D. Priority  

4. Which of the following are ACL rule fields?
   A. Rule number  
   B. Remote subnet  
   C. Endpoint  
   D. Permit/deny  

**Objective 1.3: Design Azure Compute**

You can run both Windows and Linux VMs on Azure to host your workloads. You can provision a new VM easily on Azure at any time so that you can get your workload up and running without spending the time and money to purchase and maintain any hardware. After the VM is created, you are responsible for maintenance tasks such as configuring and applying software patches.

To provide the maximum flexibility in workload hosting, Azure provides a rich image gallery with both Windows-based and Linux-based images. It also provides several different series of VMs with different amounts of memory and processor power to best fit your workloads. Furthermore, Azure supports virtual extensions with which you can customize the standard images for your project needs.
Selecting VM sizes

The easiest way to create a VM is to use the management portal. You can use either the current portal (https://manage.windowsazure.com) or the new Azure Preview Management Portal (https://portal.azure.com). The following steps use the new portal.

2. In the lower-left corner of the screen that opens, click the New icon, and then, in the center pane, select Compute, as shown in Figure 1-8. (As of this writing, the portal is still in preview, so the exact layout and naming may change.)
NOTE  ITEMS IN THE COMPUTE CATEGORY

Items in the Compute category are not just VMs. For instance, SharePoint Server Farm is composed of a group of related VMs. A group of related Azure resources is defined in a JSON-based Azure Template. An Azure Template captures the entire infrastructure of an application scenario so that you can easily deploy the entire infrastructure consistently for different purposes, such as testing, staging, and production. Azure Template is briefly introduced at the end of this objective.

In the lower-left corner of Figure 1-8, at the bottom of the list, is the option for the Azure Marketplace. This Marketplace provides thousands of first-party and third-party templates for you to deploy necessary Azure resources to support various typical workloads.

3. In this exercise, you’ll create a new Windows Server 2012 R2 VM, which happens to be the first item in the list. If the VM image is not listed, click Azure Marketplace, then click Everything, and then type in a search keyword to locate the image.

4. On the Create VM blade (the UI panes on the new Azure Portal are called blades). Type a Host Name, a User Name, and a Password, as demonstrated in Figure 1-9.

![Figure 1-9 Choosing price tier](image)

The Host Name will become the host name of your VM. Recall from Objective 1.2 that a Cloud Service with the same name will be automatically created as a container of the VM. The VM is also placed into a logical group called a Resource Group. Resource Groups are discussed when Azure Template is introduced later in this objective. The user name and the password becomes the credential of your local administrator.
5. Click Pricing Tier, which opens a new blade where you can choose from a variety of configurations (to see the complete list, click the View All link). Click a VM size that you want to use, and then click the Select button to return to the previous blade.

6. Optionally, click Optional Configuration to examine default settings and to make changes as needed. For example, you can choose to join the VM to a virtual network or create public endpoints using this blade.

7. Back on the Create VM blade, scroll down to examine other options, such as which Azure subscription to use and the region to which the VM is to be deployed. Make changes as needed.

8. Leave the Add To Starboard option selected, and then click the Create button to create the VM. After the machine is created, you’ll see a new icon on your start board (the customizable home page of the new Preview Management Portal), which provides direct access to the VM.

9. Click the icon to open the VM blade. Click the Connect icon to establish a remote desktop connection to the VM. You’ll be asked to sign in. Sign in using the credentials you entered at step 4. You’ll also see a certificate warning. Click Yes to continue.

10. After the connection is established, you can manage the VM just as if you were managing any servers through remote desktop.

Choosing pricing tiers and machine series

Azure provides two pricing tiers: Basic and Standard. Basic tier is most suitable for development, tests, and simple production workloads. It doesn’t have features such as load balancing and autoscaling. And, there are fewer VM sizes from which to choose. On the other hand, the Standard tier provides a wide range of VM sizes with features such as load balancing and autoscaling to support production workloads.

Azure organizes VM sizes into machine series—A-series, D-series, DS-series, and G-series. Only a part of A-series is available to the Basic tier. All series are available for the Standard tier. Following is a description of each series:

- **A-series** A-series VMs are designed for generic workloads. Table 1-6 lists all available sizes in the A-series. A0 to A4 sizes are available to both the Basic tier and the Standard tier. Each VM has an operating system (OS) drive and a temporary drive. The OS drives are persistent, but the temporary drives are transient. You can attach 1-TB data drives to your VMs, as well. Each has a maximum Input/Output Operations Per Second (IOPS) of 300 for the Basic tier, and 500 for the Standard tier. With more drives, you gain more overall IOPS with parallel IO operations. Among A-series sizes, A8 through A11 are designed for high-performance computing, which is discussed in Chapter 4.
TABLE 1-6  A-series VM sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>CPU cores</th>
<th>Memory</th>
<th>OS drive size (GB)/ temporary drive size (GB)</th>
<th>Maximum number of data drives</th>
<th>Maximum IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>1</td>
<td>768 MB</td>
<td>1,023/20</td>
<td>1</td>
<td>1X300/1X500</td>
</tr>
<tr>
<td>A1</td>
<td>1</td>
<td>1.75 GB</td>
<td>1,023/40</td>
<td>2</td>
<td>2X300/2X500</td>
</tr>
<tr>
<td>A2</td>
<td>2</td>
<td>3.5 GB</td>
<td>1,023/60</td>
<td>4</td>
<td>4X300/4X500</td>
</tr>
<tr>
<td>A3</td>
<td>4</td>
<td>7 GB</td>
<td>1,023/120</td>
<td>8</td>
<td>8X300/8X500</td>
</tr>
<tr>
<td>A4</td>
<td>8</td>
<td>14 GB</td>
<td>1,023/240</td>
<td>16</td>
<td>16X300/16X500</td>
</tr>
<tr>
<td>A5</td>
<td>2</td>
<td>14 GB</td>
<td>1,023/135</td>
<td>4</td>
<td>4X500</td>
</tr>
<tr>
<td>A6</td>
<td>4</td>
<td>28 GB</td>
<td>1,023/285</td>
<td>8</td>
<td>8X500</td>
</tr>
<tr>
<td>A7</td>
<td>8</td>
<td>56 GB</td>
<td>1,023/605</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>A8</td>
<td>8</td>
<td>56 GB</td>
<td>1,023/382</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>A9</td>
<td>16</td>
<td>112 GB</td>
<td>1,023/382</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>A10</td>
<td>8</td>
<td>56 GB</td>
<td>1,023/382</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>A11</td>
<td>16</td>
<td>112 GB</td>
<td>1,023/382</td>
<td>16</td>
<td>16X500</td>
</tr>
</tbody>
</table>

**NOTE ABOUT TEMPORARY DISKS**

Both OS drives and data drives are virtual hard drives (VHDs) stored in Azure Blob Storage. Their data is automatically duplicated three times for reliability. However, the temporary drives reside on the hosting servers. If the host fails, your VM will be moved to a healthy host, but not the temporary drive. In this case, you’ll lose all temporary data. By default, the temporary drive is mounted as drive D on a Windows system, and /dev/sdb1 on a Linux system.

- **D-series** This series of VMs is designed for workloads with high processing power and high-performance temporary drives. D-series VMs use solid-state drives (SSDs) for temporary storage, providing much faster IO operations compared to what traditional hard drives provide. Table 1-7 lists all available sizes in the D-series.
### TABLE 1-7  D-series VM sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>CPU cores</th>
<th>Memory (GB)</th>
<th>OS drive size (GB)/temporary drive size (GB)</th>
<th>Maximum number of data drives</th>
<th>Maximum IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard_D1</td>
<td>1</td>
<td>3.5</td>
<td>1,023/50 (SSD)</td>
<td>2</td>
<td>2X500</td>
</tr>
<tr>
<td>Standard_D2</td>
<td>2</td>
<td>7</td>
<td>1,023/100 (SSD)</td>
<td>4</td>
<td>4X500</td>
</tr>
<tr>
<td>Standard_D3</td>
<td>4</td>
<td>14</td>
<td>1,023/200 (SSD)</td>
<td>8</td>
<td>8X500</td>
</tr>
<tr>
<td>Standard_D4</td>
<td>8</td>
<td>28</td>
<td>1,023/400 (SSD)</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>Standard_D11</td>
<td>2</td>
<td>14</td>
<td>1,023/100 (SSD)</td>
<td>4</td>
<td>4X500</td>
</tr>
<tr>
<td>Standard_D12</td>
<td>4</td>
<td>28</td>
<td>1,023/200 (SSD)</td>
<td>8</td>
<td>8X500</td>
</tr>
<tr>
<td>Standard_D13</td>
<td>8</td>
<td>56</td>
<td>1,023/400 (SSD)</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>Standard_D14</td>
<td>16</td>
<td>112</td>
<td>1,023/800 (SSD)</td>
<td>32</td>
<td>32X500</td>
</tr>
</tbody>
</table>

EXAM TIP

A0 to A4 are called by different names in the Basic tier and the Standard tier. In the Basic tier, they are referred as Basic_A0 to Basic_A4, whereas in the Standard series, each of the sizes has its own corresponding names, which are used in scripts and API calls. You should know how these names mapped to VM sizes:

- A0: extra small
- A1: small
- A2: medium
- A3: large
- A4: extra large

**DS-series** DS-Series VMs are designed for high I/O workloads. They use SSDs for both VM drives and a local drive cache. Table 1-8 lists all DS-series sizes.

### TABLE 1-8  DS-series VM sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>CPU cores</th>
<th>Memory (GB)</th>
<th>OS drive size (GB)/local drive size (GB)</th>
<th>Maximum number of data drives</th>
<th>Cache Size (GB)</th>
<th>Maximum IOPS/bandwidth (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard_DS1</td>
<td>1</td>
<td>3.5</td>
<td>1,023/7 (SSD)</td>
<td>2</td>
<td>43</td>
<td>3,200/32</td>
</tr>
<tr>
<td>Standard_DS2</td>
<td>2</td>
<td>7</td>
<td>1,023/14 (SSD)</td>
<td>4</td>
<td>86</td>
<td>6,400/64</td>
</tr>
<tr>
<td>Standard_DS3</td>
<td>4</td>
<td>14</td>
<td>1023/28 (SSD)</td>
<td>8</td>
<td>172</td>
<td>12,800/128</td>
</tr>
<tr>
<td>Standard_DS4</td>
<td>8</td>
<td>28</td>
<td>1,023/56 (SSD)</td>
<td>16</td>
<td>344</td>
<td>25,600/256</td>
</tr>
<tr>
<td>Standard_DS11</td>
<td>2</td>
<td>14</td>
<td>1,023/28 (SSD)</td>
<td>4</td>
<td>72</td>
<td>6,400/64</td>
</tr>
<tr>
<td>Standard_DS12</td>
<td>4</td>
<td>28</td>
<td>1,023/56 (SSD)</td>
<td>8</td>
<td>144</td>
<td>12,800/128</td>
</tr>
<tr>
<td>Standard_DS13</td>
<td>8</td>
<td>56</td>
<td>1,023/112 (SSD)</td>
<td>16</td>
<td>288</td>
<td>25,600/256</td>
</tr>
<tr>
<td>Standard_DS14</td>
<td>16</td>
<td>112</td>
<td>1,023/224 (SSD)</td>
<td>32</td>
<td>576</td>
<td>50,000/512</td>
</tr>
</tbody>
</table>
NOTE AVAILABILITY OF DS-SERIES

DS-series requires Premium storage (see Chapter 3), which is currently available in certain regions (West US, East US 2, and West Europe).

- **G-series**  This series of VMs is one of the biggest on cloud with Xeon E5 V3 family processors. Table 1-9 lists all available sizes in the G-series.

<table>
<thead>
<tr>
<th>Size</th>
<th>CPU cores</th>
<th>Memory (GB)</th>
<th>OS drive size (GB)/local drive size (GB)</th>
<th>Maximum number of data drives</th>
<th>MAX IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard_G1</td>
<td>2</td>
<td>28</td>
<td>1,023/384 (SSD)</td>
<td>4</td>
<td>4X500</td>
</tr>
<tr>
<td>Standard_G2</td>
<td>4</td>
<td>56</td>
<td>1,023/768 (SSD)</td>
<td>8</td>
<td>8X500</td>
</tr>
<tr>
<td>Standard_G3</td>
<td>8</td>
<td>112</td>
<td>1,023/1,536 (SSD)</td>
<td>16</td>
<td>16X500</td>
</tr>
<tr>
<td>Standard_G4</td>
<td>16</td>
<td>224</td>
<td>1,023/3,072 (SSD)</td>
<td>32</td>
<td>32X500</td>
</tr>
<tr>
<td>Standard_G5</td>
<td>32</td>
<td>448</td>
<td>1,023/6,144 (SSD)</td>
<td>64</td>
<td>64X500</td>
</tr>
</tbody>
</table>

NOTE INCREASED OS DRIVE SIZES

The maximum capacity of OS drives used to be 127 GB. Since March 2015, this limit has been increased to 1,023 GB.

Using data disks

As previously mentioned, temporary drives are transient and you should not use them to maintain permanent data. If your application needs local storage to keep permanent data, you should use data drives. The Tables 1-6 through 1-9 show that for each VM size you can attach a number of data drives. You can attach both empty data drives and data drives with data to a VM. To attach a data drive, go to the Settings blade of your VM, click Disks, and then select either Attach New to create a new data drive, or Attach Existing to attach an existing data drive. Figure 1-10 shows demonstrates attaching a new data drive to a VM using the Preview Management Portal.
DRIVES, IMAGES, AND VHD FILES

Both OS drives and data drives are VHD files stored on your Storage account. A VM is created from a VM image, which might correspond to one or more VHD files. Azure provides a number of standard OS images, and you can upload or capture your own images and use those images to create VMs. When you capture an image of a VM, you can capture both the OS drives and data drives so that you can replicate the entire VM elsewhere as needed. You’ll learn about capturing VM images later in this objective.

After a new data drive is attached to a VM, you need to initialize it before you can use it. For Windows-based VMs, you can use the Disk Manager tool in Server Manager to initialize the drive, and then create a simple volume on it, or a striped volume across multiple drives. For Linux-based VMs, you need to use a series of commands such as `fdisk`, `mkfs`, `mount`, and `blkid` to initialize and mount the drive.

You can choose a host caching preference—None, Read Only, or Read/Write—for each data drive. The default settings usually work fine, unless you are hosting database workloads or other workloads that are sensitive to small I/O performance differences. For a particular workload, the best way to determine which preference to use is to perform some I/O benchmark tests.

Generally speaking, using striped drives usually yields better performance for I/O-heavy applications. However, you should avoid using geo-replicated storage accounts for your striped volumes because data loss can occur when recovering from a storage outage (for more information, go to https://msdn.microsoft.com/en-us/library/azure/dn790303.aspx).
Managing images

There are three sources for Azure VM: the Azure VM gallery, VM Depot, and custom images. You can use these images as foundations to create, deploy, and replicate your application run-time environments consistently for different purposes such as testing, staging, and production.

- **VM gallery**  The Azure VM gallery offers hundreds of VM images from Microsoft, partners, and the community at large. You can find recent Windows and Linux OS images as well as images with specific applications, such as SQL Server, Oracle Database, and SAP HANA. MSDN subscribers also have exclusive access to some images such Windows 7 and Windows 8.1. For a complete list of the images, go to [http://azure.microsoft.com/en-us/marketplace/virtual-machines/](http://azure.microsoft.com/en-us/marketplace/virtual-machines/).

- **VM Depot** The VM Depot ([https://vmdepot.msopentech.com/List/Index](https://vmdepot.msopentech.com/List/Index)) is an open-source community for Linux and FreeBSD images. You can find an increasing number of images with various popular open-source solutions such as Docker, Tomcat, and Juju.

- **Custom images** You can capture images of your VMs and then reuse these images as templates to deploy more VMs.

Capturing custom images

You can capture two types of images: generalized or specialized.

A generalized image doesn’t contain computer or user-specific settings. These images are ideal for use as standard templates to rollout preconfigured VMs to different customers or users. Before you can capture a generalized image, you need to run the System Preparation (Sysprep) tool in Windows, or use the `waagent –deprovision` command in Linux. All the OS images you see in the VM gallery are generalized. Before you can capture a generalized image, you need to shut down the VM. After the VM is captured as an image, the original VM is automatically deleted.

Specialized images, conversely, retain all user settings. You can think of specialized images as snapshots of your VMs. These images are ideal for creating checkpoints of an environment so that it can be restored to a previously known good state. You don’t need to shut down a VM before you capture specialized images. Also, the original VM is unaffected after the images are captured. If a VM is running when an image is captured, the image is in crash-consistent state. If application consistency or cross-drive capture is needed, it’s recommended to shut down the VM before capturing the image.

To capture an image, on the Virtual Machine blade, click the Capture button, as shown in Figure 1-11.
Using custom images

You can use your custom images to create new VMs just as you would use standard images. If you use a specialized image, you skip the user provisioning step because the image is already provisioned. When a new VM is created, the original VHD files are copied so that the original VHD files are not affected.

As of this writing, there's no easy way to use custom images on the new Preview Management Portal. However, with the full management portal, you can use custom images by clicking the My Images link on the Choose An Image page of the Create A Virtual Machine Wizard, as illustrated in Figure 1-12.

Alternatively, you can use Azure PowerShell to create a new VM by using a custom image. For example, to create a new VM from the custom image myVMImage, use the following command:

New-AzureQuickVM -Windows -Location "West US" -ServiceName "examService" -Name "examVM" -InstanceSize "Medium" -ImageName "myVMImage" -AdminUsername "admin" -Password "sh@ng3!" -WaitForBoot
Managing VM states

Custom images provide basic supports for deploying workloads consistently across different environments. However, custom images have some undesirable characteristics. First, it’s difficult to revise a custom image. To make any changes, you need to provision the image as a new VM, customize it, and then recapture it. Second, it’s also difficult to track what has been changed on an image because of the manual customizations. Third, rolling out a new version is difficult, as well. To deploy a new image version, the VM needs to be re-created, making upgrade a lengthy and complex process. What you need are more light-weight, traceable, agile, and scalable state management solutions. This section discusses a number of technologies that enable efficient VM state managements.

VM extension

When you provision a new VM, a light-weight Azure Virtual Machine Agent (VM Agent) is installed on the VM by default. VM Agent is responsible for installing, configuring, and managing Azure VM Extensions (VM Extensions). VM Extensions are first-party or third-party components that you can dynamically apply to VMs. These extensions make it possible for you to dynamically customize VMs to satisfy your application, configuration, and compliance needs. For example, you can deploy the McAfee Endpoint Security extension to your VMs by enabling the \textit{McAfeeEndpointSecurity} extension.

You can use Azure PowerShell cmdlet \texttt{Get-AzureVMAvailableExtension} to list currently available extensions. Listing 1-1 shows a sample of the cmdlet.

\textbf{LISTING 1-1 Listing available VM extensions}

\begin{verbatim}
PS C:\> Get-AzureVMAvailableExtension | Format-Table -Wrap -AutoSize -Property
ExtensionName, Description
ExtensionName Description
------------- -----------
VS14CTPDebugger     Remote Debugger for Visual Studio 2015
ChefClient          Chef Extension that sets up chef-client on VM
LinuxChefClient     Chef Extension that sets up chef-client on VM
DockerExtension     Docker Extension
DSC                 PowerShell DSC (Desired State Configuration) Extension
CustomScriptForLinux Microsoft Azure Custom Script Extension for Linux IaaS
BGInfo              Windows Azure BGInfo Extension for IaaS
CustomScriptExtension Windows Azure Script Handler Extension for IaaS
VMAccessAgent       Windows Azure Json VMAccess Extension for IaaS
...
\end{verbatim}

\textbf{NOTE VM AGENT OPT-OUT}

When creating a VM, you can choose not to install VM agent. You can install VM Agent to an existing VM; however, when a VM agent is installed, removing it is not a supported scenario. You can, of course, physically remove the agent, but the exact behavior after removal is unsupported.
Custom Script Extension and DSC

Custom Script Extension downloads and runs scripts you’ve prepared on an Azure Blob storage container. You can upload Azure PowerShell scripts or Linux Shell scripts, along with any required files, to a storage container, and then instruct Custom Script Extension to download and run the scripts. The following code snippet shows a sample Azure CLI command to use the Custom Script Extension for Linux (CustomScriptForLinux) to download and run a mongod.sh shell script:

```
azure vm extension set -t '{"storageAccountName": "[storage account]", "storageAccountKey": ""}" -i '{"fileUris": ["http://[storage account].blob.core.windows.net/scripts/mongodb.sh"], "commandToExecute": "sh mongodb.sh"]' [vm name] CustomScriptForLinux
```

Using scripts to manage VM states overcomes the shortcomings of managing them with images. Scripts are easier to change and you can apply them faster. And an added benefit is that you can trace all changes easily by using source repositories.

However, writing a script to build up a VM toward a target state is not easy. For each of the requirement components, you’ll need to check if the component already exists and if it is configured in the desired way. You’ll also need to deal with the details of acquiring, installing, and configuring various components to support your workloads. Windows PowerShell Desired State Configuration (DSC) takes a different approach. Instead of describing steps of how the VM state should be built up, you simply describe what the desired final state is with DSC. Then, DSC ensures that the final state is reached. The following is a sample DSC script that verifies the target VM has IIS with ASP.NET 4.5 installed:

```
Configuration DemoWebsite
{
    param ($MachineName)
    Node $MachineName
    {
        #Install the IIS Role
        WindowsFeature IIS
        {
            Ensure = "Present"
            Name = "Web-Server"
        }
        #Install ASP.NET 4.5
        WindowsFeature ASP
        {
            Ensure = "Present"
            Name = "Web-Asp-Net45"
        }
    }
}
```
State management at scale

For larger deployments, you often need to ensure consistent states across a large number of VMs. You also need to periodically check VM states so they don’t drift from the desired parameters. An automated state management solution such as Chef and Puppet can save you from having to carry out such repetitive and error-prone tasks.

For both Chef and Puppet, you write cookbooks that you can then apply to a large number of VMs. Each cookbook contains a number of “recipes” or “modules” for various tasks, such as installing software packages, making configuration changes, and copying files. They both facilitate community contributions (Puppet Forge and Chef Supermarket) so that you can accomplish common configuration tasks easily. For example, to get a Puppet module that installs and configures Redis, you can use Puppet tool to pull down the corresponding module from Puppet Forge:

```
puppet module install evenup-redis
```

Both Chef and Puppet install agents on your VMs. These agents monitor your VM states and periodically check with a central server to download and apply updated cookbooks. Azure provides VM extensions that bootstrap Chef or Puppet agents on your VMs. Furthermore, Azure also provides VM images that assist you in provisioning Chef and Puppet servers. Chef also supports a hosted server at https://manage.chef.io.

Managing VM states is only part of the problem of managing application run-time environments in the cloud. Your applications often depend on external services. How do you ensure that these external services remain in desired states? The solution is Azure Automation. With Automation, you can monitor events in VMs as well as external services such as Azure App Service Web Apps, Azure Storage, and Azure SQL Server. Then, workflows can be triggered in response to these events.

Automation’s cookbooks, called runbooks, are implemented as Azure PowerShell Workflows. To help you to author these runbooks, Azure has created an Azure Automation Runbook Gallery where you can download and share reusable runbooks. Figure 1-13 shows how you can create a new runbook based on existing runbooks in the gallery.
Capturing infrastructure as code

Traditionally, development and operations are two distinct departments in an Independent Software Vendor (ISV). Developers concern themselves with writing applications, and the folks in operations are concerned with keeping the applications running. However, for an application to function correctly, there are always explicit or implicit requirements regarding how the supporting infrastructure is configured. Unfortunately, such requirements are often lost during communication, which leads to many problems such as service outages because of misconfigurations, frictions between development and operations, and difficulties in recreating and diagnosing issues. All these problems are unacceptable in an Agile environment.

In an Agile ISV, the boundary between development and operations is shifting. The developers are required to provide consistently deployable applications instead of just application code; thus, the deployment process can be automated to rollout fixes and upgrades quickly. This shift changed the definition of application. An application is no longer just code. Instead, an application is made up of both application code and explicit, executable description of its infrastructural requirements. For the lack of better terminology, such descriptions can be called infrastructure code. The name has two meanings. First, “infrastructure” indicates that it’s not business logics but instructions to configure the application runtime. Second “code” indicates that it’s not subject to human interpretation but can be consistently applied by an automation system.

Infrastructure code is explicit and traceable, and it makes an application consistently deployable. Consistently deployable applications are one of the key enabling technologies in the DevOps movement. The essence of DevOps is to reduce friction so that software lifecycles...
can run smoother and faster, allowing continuous improvements and innovations. Consistently deployable applications can be automatically deployed and upgraded regularly across multiple environments. This means faster and more frequent deployments, reduced confusion across different teams, and increased agility in the overall engineering process.

**Azure Resource Template**

Azure Resource Templates are JSON files that capture infrastructure as code. You can capture all the Azure resources your application needs in a single JSON document that you can consistently deploy to different environments. All resources defined in an Azure Resource Template are provisioned within a Resource Group, which is a logical group for managing related Azure resources.

**NOTE  SUPPORTING ALL AZURE RESOURCE TYPES**

Supports of all Azure resource types are added gradually over time. Each Azure resource type is implemented as a Resource Provider that can be plugged into Azure Resource Manager, the service that governs resource creation. At this point, there are only a number of Azure resource types that are supported. Eventually, though, all Azure resource types are expected to be supported.

You can write an Azure Resource Template from scratch using any text editor. You can also download a template from an Azure template gallery by using Azure PowerShell:

1. In Azure PowerShell, switch to Azure Resource Manager mode:
   
   ```
   Switch-AzureMode AzureResourceManager
   ```

2. Use the `Get-AzureResourceGroupGalleryTemplate` cmdlet to list gallery templates. The command returns a large list. You can use the `Publisher` parameter to constrain the results to a specific publisher:
   
   ```
   Get-AzureResourceGroupGalleryTemplate -Publisher Microsoft
   ```

3. Save and edit the template of interest:
   
   ```
   Save-AzureResourceGroupGalleryTemplate -Identity Microsoft.JavaCoffeeShop.0.1.3-preview -Path C:\Templates\JavaCoffeeShop.json
   ```

4. At the top of the file, an Azure Resource Template contains a schema declaration (Figure 1-14). This consists of a content version number and a “resources” group, which contains resource definitions.

   ```
   
   ```

   **FIGURE 1-14 Sample Azure Template**
Optionally, you can also define parameters, variables, tags, and outputs. A complete introduction of the template language is beyond the scope of this book. You can use the `Test-AzureResourceGroupTemplate` cmdlet to validate your template at any time. You need an actual Resource Group in order to use the cmdlet. However, creating a Resource Group is easy:

```powershell
New-AzureResourceGroup -Name [resource group name]
```

5. Supply the resource group name to the command along with other required parameters, and then validate if your template is ready to be deployed.

To deploy a template, use the `New-AzureResourceGroupDeployment` cmdlet:

```powershell
```

An Azure Resource Template captures the entire topology of all Azure resources required by your application. And, you can deploy it with a single Azure PowerShell command. This capacity greatly simplifies resource management of complex applications, especially service-oriented architecture (SOA) applications that often have many dependencies on hosted services.

**Containerization**

In the past few years, container technologies such as Docker have gained great popularity in the industry. Container technologies make it possible for you to consistently deploy applications by packaging them and all their required resources together as a self-contained unit. You can build a container manually, or it can be fully described by metadata and scripts. This way, you can manage containers just as source code. You can check them in to a repository, manage their versions, and reconcile their differences just as how you would manage source code. In addition, containers have some other characteristics that make them a favorable choice for hosting workloads on cloud, which are described in the sections that follow.

**AGILITY**

Compared to VMs, containers are much more light weight because containers use process isolation and file system virtualization to provide process-level isolations among containers. Containers running on the same VM share the same system core so that the system core is not packaged as part of the container. Because starting a new container instance is essentially the same as starting a new process, you can start containers quickly—usually in time frames less than a second. The fast-start time makes containers ideal for the cases such as dynamic scaling and fast failover.
COMPUTE DENSITY
Because container instances are just processes, you can run a large number of container instances on a single physical server or VM. This means that by using containers, you can achieve much higher compute density in comparison to using VMs. A higher compute density means that you can provide cheaper and more agile compute services to your customers. For example, you can use a small number of VMs to host a large number of occasionally accessed websites, thus keeping prices competitive. And you can schedule a larger number of time-insensitive batch jobs.

DECOUPLE COMPUTE AND RESOURCE
Another major benefit of using containers is that the workloads running in them are not bound to specific physical servers or VMs. Traditionally, after a workload is deployed, it's pretty much tied to the server where it's deployed. If the workload is to be moved to another server, the new one needs to be repurposed for the new workload, which usually means the entire server needs to be rebuilt to play its new role in the datacenter. With containers, servers are no longer assigned with specific roles. Instead, they form a cluster of CPUs, memory, and disks within which workloads can roam almost freely. This is a fundamental transformation in how the datacenter is viewed and managed.

Container orchestration
There are many container orchestration solutions on the market that provide container clustering, such as Docker Swarm, CoreOS Fleet, Deis, and Mesosphere. Orchestrated containers form the foundation of container-based PaaS offerings by providing services such as coordinated deployments, load balancing, and automated failover.

EXAM TIP
Container technology has gained considerable momentum in the past few years. New capabilities, new services, and new companies are emerging rapidly and the landscape is changing continually. For example, there are many variations in capabilities of different orchestration offerings. At this point, the container should not be a focus of the test, so you shouldn't spend a lot of energy to chase new developments in the field. However it's very important to understand the benefits of containers because they will become increasingly important in future tests.

Orchestrated containers provide an ideal hosting environment for applications that use Microservices architecture. You can package each service instance in its own corresponding container. You can join multiple containers together to form a replica set for the service. You can automate container cluster provisioning by using a combination of Azure Resource Template, VM Extensions, Custom Script Extension, and scripts. The template describes the cluster topology, and VM extensions perform on-machine configurations. Finally, automated scripts in containers themselves can perform container-based configurations.
Scaling applications on VMs

In Azure, you can configure applications to *scale-up* or *scale-out.*

Scaling-up refers to increasing the compute power of the hosting nodes. In an on-premises datacenter, scaling up means to increase the capacity of the servers by increasing memory, processing power, or drive spaces. Scaling-up is constrained by the number of hardware upgrades you can fit into the physical machines. In the cloud, scaling-up means to choose a bigger VM size. In this case, scaling-up is constrained by what VM sizes are available.

Scaling-out takes a different approach. Instead of trying to increase the compute power of existing nodes, scaling-out brings in more hosting nodes to share the workload. There’s no theoretical limit to how much you can scale-out—you can add as many nodes as needed. This makes it possible for an application to be scaled to very high capacity that is often hard to achieve with scaling-up. Scaling-out is a preferable scaling method for cloud applications.

The rest of this section will focus on scaling out.

Load balancing

When you scale-out an application, the workload needs to be distributed among the participating instances. This is done by load balancing. (Load-balanced endpoints were introduced earlier in this chapter.) The application workload is distributed among the participating instances by the Azure public-facing load-balancer in this case.

However, for multitiered applications, you often need to scale-out middle tiers that aren’t directly accessible from the Internet. For instance, you might have a website as the presentation layer, and a number of VMs as the business layer. You usually don’t want to expose the business layer, and thus you made it accessible only by the presentation layer. How would you scale the business layer without a public-facing load balancer? To solve this problem, Azure introduces Internal Load Balancers (ILB). ILBs provide load balancing among VMs residing in a Cloud Service or a regional virtual network.

The ILBs are not publically accessible. Instead, you can access them only by other roles in the same Cloud Services, or other VMs within the same virtual network. ILB provides an ideal solution for scaling a protected middle tier without exposing the layer to the public. Figure 1-15 shows a tiered application that uses both a public-facing load balancer and an internal load balancer. With this deployment, end users access the presentation layer through Secure Sockets Layer (SSL). The requests are distributed to the presentation layer VMs by Azure Load Balancer. Then, the presentation layer accesses the database servers through an internal load balancer.
As mentioned earlier, you can define custom health probes when you define a load-balanced set. You can configure your VMs to respond to health probes from the load balancer via either TCP or HTTP. If a VM fails to respond to a given number of probes, it is considered unhealthy and taken out of the load balancer. The load balancer will keep probing all of the VMs (including the unhealthy ones) so that when the failed VM is recovered, it will automatically be rejoined to the balanced set. You can use this feature to temporarily take a VM off the load balancer for maintenance by forcing a false response to probe signals.

**Autoscale**

With Azure, you can scale your VMs manually in a Cloud Service. In addition, you can also set up autoscale rules to dynamically adjust the system capacity in response to average CPU usage or number of messages in a queue.
To use autoscaling, you need to add the VMs to an Availability Set. Availability Sets are discussed in Chapter 4. At the moment, you can consider an Availability Set as a group of VMs for which Azure attempts to keep at least one VM running at any given time. Figure 1-16 shows a sample autoscaling policy on the management portal.

Let’s go through the above policy item by item.

- **Edit Scale Settings For Schedule** You can specify different scaling policies for different times of the day, different days of the week, and specific date ranges. For example, if you are running an e-commerce site that expects spikes in traffic during weekends, you can set up a more aggressive scaling policy to ensure that the performance of the system under heavier loads during those periods.

- **Scale By Metric** You can choose None, CPU, or Queue. An autoscaling policy without a scale metric is for scheduled scaling scenarios. For the latter two options, Azure monitors the performance of your VMs and adjusts the number of instances accordingly to ensure that the metric falls into the specified range.

- **Instance Range** The Instance Range specifies the lower and upper boundaries of scaling. The lower boundary makes certain that the system maintains a minimum capacity, even if the system is idle. The upper boundary controls the cost limit of your deployment. Each VM instance has its associated cost. You want to set up an appropriate upper limit so that you don’t exceed your budget.
- **Target CPU**  The Target CPU specifies the desired range of the specific metric. If the value exceeds the upper limit, scaling up (in this case, a more precise term would be “scaling out”) will be triggered. If the value falls below the lower limit, scaling down (again, in this case a more precise term would be “scaling in”) will be triggered. Please note that the autoscaling system doesn’t respond to every metric value changes. Instead, it makes decisions based on the average value in the past hour.

**NOTE AUTOSCALING AFTER THE FIRST HOUR**

Because autoscaling uses the average value of the past hour, it’s not triggered as frequently as you might have expected. This is a very common point of confusion for many users who set up the autoscaling policy for the first time.

- **Scale Up By**  You can specify how fast the system is to be scaled-out by specifying scaling steps and delays between scaling actions.

- **Scale Down By**  You can control how the system is scaled down. Depending on how your workload pattern changes, you might want to set an aggressive scale-down policy to de-provision the resources quickly after busy hours to reduce your costs.

---

**Thought experiment**

**Lift and ship**

In this thought experiment, apply what you’ve learned about this objective. You can find answers to these questions in the “Answers” section at the end of this chapter.

When it comes to adopting the cloud, many enterprises would consider lifting and shipping existing applications to Azure VMs as the starting point. After all, if an application runs well on a local server, the chances are good that it will run well on a VM in Azure, won’t it? However, there are often more things to consider than just deploying the applications to a VM, such as reliability, availability, security, and performance.

With this in mind, answer the following questions:

1. What challenges do you think you need to prepare for?
2. What are the next steps after an application has been moved to Azure?
Objective summary

- Azure supports various VM sizes and a gallery of both Linux images and Windows images.
- You can automate VM state management with Azure Automation and third-party solutions such as Chef and Puppet.
- VM Extension and Azure PowerShell DSC automates on-machine configuration tasks.
- DevOps requires infrastructure to be captured as code. With DevOps, an application consists of both application code and infrastructure code so that the application can be deployed consistently and rapidly across different environments.
- Azure Resource Template captures the entire topology of your application as code, which you can manage just as you do application source code. Resource Templates are JSON files that you can edit using any text editors.
- Containerization facilitates agility, high compute density, and decoupling of workloads and VMs. It transforms the datacenter from VMs with roles to resource pools with mobilized workloads.
- You can use autoscale to adjust your compute capacity to achieve balance between cost and customer satisfaction.

Objective review

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of this chapter.

1. What VM series should you consider if you want host applications that require high-performance IO for persisted data?
   A. A-series
   B. D-series
   C. DS-series
   D. G-series

2. How many data drives can you attach to a Standard_G5 VM (the biggest size in the series)?
   A. 8
   B. 16
   C. 32
   D. 64
3. What's the format of an Azure Resource Template?
   A. JSON  
   B. XML  
   C. YAML  
   D. PowerShell

4. Which of the following technologies can help you to manage consistent states of VMs at scale?
   A. Custom Script Extension  
   B. Chef or Puppet  
   C. Azure Automation  
   D. Containerization

**Objective 1.4: Describe Azure virtual private network (VPN) and ExpressRoute architecture and design**

Microsoft realizes that for many of its existing enterprise customers, migration to cloud will be a long process that might take years or event decades. In fact, for some of these customers, a complete migration might never be feasible. To ensure smooth cloud transitions, Azure provides a pathway for enterprises to adopt cloud at their own pace. This means that for the foreseeable future, many enterprises will be operating *hybrid* solutions that have components running both on-premises and in the cloud. Thus, reliable, secure, and efficient connectivity between on-premises datacenters and cloud becomes a necessity. This objective discusses two of the connectivity options: Azure Virtual Network and Azure ExpressRoute. Then, we briefly introduce some of the other hybrid solution options.

This section covers the following topics:
- Designing hybrid solutions with Virtual Network and ExpressRoute
- Understanding other hybrid solution options

**Designing hybrid solutions with Virtual Network and ExpressRoute**

Virtual Network offers several types of hybrid connections that bridge resources located at different facilities. You can choose one or several connection options that best suit your requirements. Note that this objective does not focus on detailed steps of setting up the connections. Instead, it describes the steps in general and then focuses on how each connection type suits different scenarios.
Point-to-Site VPN

Point-to-Site VPN is the simplest hybrid connection by which you can securely connect your local computer to an Azure virtual network. No specific VPN devices are needed in this case. Instead, you install a Windows VPN client through which you can connect to any VMs and Cloud Services within the virtual network. Figure 1-17 shows the topology of a Point-to-Site VPN.

FIGURE 1-17  Point-to-site connectivity

Establishing a point-to-site connection involves several steps:

1. Specify an IP address range. When your VPN clients connect, they will receive IP addresses from this range. You need to ensure that this range doesn’t overlap with IP ranges within your on-premises network.

2. Add a gateway subnet.

3. Create a dynamic routing gateway.

   You can choose between a standard gateway, which gives you about 80 Mbps and 10 S2S tunnels, and a high-performance gateway, which gives you about 200 Mbps and 30 S2S tunnels.

4. Create a client certification to be used for client authentication. The client machine that makes the VPN connection needs to have the certificate installed.

5. Download the VPN client configuration package from your virtual network’s Dashboard page. When the client is installed, you’ll see a new VPN connection with the same name as your virtual network.

With Point-to-Site connection, you can connect to your VMs on Azure from anywhere. It uses Secured Socket Tunneling Protocol (SSTP), which means that you can establish the connection through firewalls and Network Address Translation (NAT). It works well to support a small mobile workforce. However, because each client PC in this case establishes a separate connection to the gateway, you are limited to the number of S2S tunnels that the gateway can support.

Point-to-Site enables scenarios such as remote administration of cloud resources, troubleshooting, monitoring, and testing. It can be applied to use cases such as remote education, mobile office, and occasional command and control. However, for bridging on-premises networks and Azure Virtual Networks, you’ll probably want to use Site-to-Site VPN.
Site-to-Site VPN

Site-to-Site VPN is designed for establishing secured connections between site offices and the cloud, or bridging on-premises networks with virtual networks on Azure. To establish a Site-to-Site VPN connection, you need a public-facing IPv4 address and a compatible VPN device, or Routing and Remote Access Service (RRAS) running on Windows Server 2012. (For a list of known compatible devices, go to https://msdn.microsoft.com/en-us/library/azure/jj156075.aspx#bkmk_KnownCompatibleVPN.) You can use either static or dynamic gateways for Site-to-Site VPN. However, if you want to use both Site-to-Site VPN and Point-to-Site VPN at the same time, you'll need a dynamic gateway. Figure 1-18 shows the topology of a Site-to-Site VPN.

Site-to-Site VPN extends your local network to the cloud. As you move your workloads gradually to the cloud, you often need the servers in the cloud and the local servers to still work together before the migration is complete. Using Site-to-Site VPN, these servers can communicate with each other as if they were on the same local network. This becomes handy when you move some domain-joined servers to the cloud but you still want to keep them on your local Active Directory.

Site-to-Site works in the other direction, as well: it brings your VMs in the cloud into your local network. You can join these servers into your local domain and apply your security policies on them. In many migration cases, moving the application servers is easier compared to moving a large amount of data. And some enterprises prefer to keep their data local for various reasons. With Site-to-Site VPN, your cloud VMs can reach back to your on-premises data. They also can be joined to Azure Load Balancer to provide high-availability services.

Although Site-to-Site connections provide reasonable reliability and throughput, some larger enterprises require much more bandwidth between their datacenters and the cloud. Moreover, because VPNs go through the public Internet, there's no SLA to guarantee the connectivity. For these enterprises, ExpressRoute is the way to go.
ExpressRoute

ExpressRoute provides private connections between your on-premises datacenters and Azure datacenters. You can achieve up to 10 Gbps bandwidth with the dedicated, secure, and reliable connections. These connections don’t go through the public Internet, and you can get connectivity SLAs from your selected service providers. If you have frequent large-volume data transfers between your on-premises datacenters and Azure, ExpressRoute provides a faster solution that in some cases is even more economical.

There are two ways to use ExpressRoute to connect to Azure. One way is to connect to Azure through an exchange provider location. The other way is to connect Azure through a network service provider. The exchange provider option provides up to 10 Gbps bandwidth. The network service provider option provides up to 1 Gbps bandwidth. In either case, Azure configures a pair of cross-connections between Azure and the provider’s infrastructure in an active-active configuration to ensure availability and resilience against failures. Figure 1-19 shows the topology of an ExpressRoute connection.

ExpressRoute’s fast and reliable connection is ideal for scenarios such as data storage access, backups, and disaster recovery. For example, you can transfer and store a large amount of data to Azure Storage service while keeping your applications running on your own datacenter. For backup and disaster recovery, ExpressRoute makes data replication faster and more reliable, improving the performance as well as the reliability of your disaster recovery strategies. Moreover, you can access other Azure-hosted services such as Office 365 by using the same private connection for fast, secure access.

**NOTE**  **AVAILABILITY OF EXPRESSROUTE TO OFFICE 365**

ExpressRoute to Office 365 connectivity is expected to be available to Office 365 customers beginning in the latter part of 2015.

When working together, many servers need frequent exchanges of data. When some of the servers are moved to the cloud, the additional latency introduced by Internet connections can have a serious impact on the performance of the overall system and sometimes render
the entire system unusable. ExpressRoute provides a fast connection between your on-premises datacenters and Azure so that you can extend your local infrastructure to the cloud without having to make significant architecture or code changes.

vNet-to-vNet VPN

Just as you can establish Site-to-Site connections between your on-premises datacenters and Azure, you also can connect two virtual networks on Azure by using a VPN connection. Figure 1-20 shows the topology of a vNet-to-vNet connection.

![vNet-to-vNet connectivity](image)

**FIGURE 1-20** vNet-to-vNet connectivity

You can use vNet-to-vNet VPN to support georedundancy and geopresence. For example, you can use vNet-to-vNet VPN to set up SQL Always On across multiple Azure regions. Figure 1-21 shows another example, which is a cross-region three-node MongoDB replica set with a primary node and a secondary node in West US, and a secondary in West Europe. The West Europe node is for disaster recovery and is not allowed to be elected as a primary.

![Cross-region MongoDB replica set](image)

**FIGURE 1-21** Cross-region MongoDB replica set

You also can use vNet-to-vNet VPN in business integration scenarios. With global corporations, business units sometimes remain independent from one another, but at the same time...
some workflows need to be integrated. Using vNet-to-vNet, resources owned by different business units can communicate with one another while maintaining isolations between the resources (refer to the earlier discussions on ACLs and NSGs). Some multitiered applications need such kind of isolations, as well. For instance, a new corporate website might need to consume services and data from multiple regional sites, which have their own virtual networks and security policies.

**Multi-site VPN**

You can use an Azure Virtual Network gateway to establish multiple Site-to-Site connections. This capability makes it possible to join multiple on-premises networks. Figure 1-22 shows the topology of a Multi-site VPN.

Using Multi-site VPN, branch offices from different geographic locations can connect with one another to exchange data and share Azure-based resources such as a common hosted services. This topology is also referred to as a *hub-and-spoke* topology, which is quite common for scenarios in which a head office connects to multiple branch offices.
Understanding other hybrid solution options

In addition to various networking solutions, Azure also provides other services and tools that help you to implement hybrid scenarios. This section provides a brief review of these services and tools in the contexts of different scenarios.

Reaching out to the cloud

In this scenario, you have some locally hosted services that you want to expose to the cloud.

- **Service Bus Relay**  With this service, you can expose your Windows Communication Foundation (WCF) services by registering a relay endpoint. Even if your service is behind a firewall and on a NAT, service consumers can still access the service via the public relay endpoint.

- **API Management**  Using Azure API Management, you can modernize, manage, protect, and monitor your existing APIs hosted either on-premises or on cloud.

Reaching back to on-premises

In this scenario, your cloud-based services need to reach back to your on-premises resources such as databases in your local datacenter. You can use Azure App Service BizTalk API Apps Hybrid Connection to connect your web applications back to any on-premises resources that use a static TCP port, such as SQL database and Web APIs. This service is introduced briefly in Chapter 4.

---

**Thought experiment**

Dealing with network latency

In this thought experiment, apply what you’ve learned about this objective. You can find answers to these questions in the “Answers” section at the end of this chapter.

When you have servers running on both on-premises and the cloud, it’s almost unavoidable that you will experience some performance degradation because of the extra network latency. When the degradation becomes unacceptable, some modifications to the code or to the architecture become necessary.

With this in mind, answer the following questions:

1. What code changes would you make to reduce latency?
2. What architecture changes would you make to reduce latency?
Objective summary

- You can use Point-to-Site connections to connect local compute to Azure Virtual Networks.
- You can use Site-to-Site connections to connect on-premises network to Azure Virtual Networks.
- You can use ExpressRoute to create a private, dedicated connection between your datacenters and Azure datacenters.
- To connect two Azure virtual networks, use vNet-to-vNet VPN.
- To connect multiple on-premises networks to the same Azure virtual network, use Multi-site VPN.
- You can use Service Bus Relay and API Management to expose local services to cloud.
- You can use BizTalk API Apps Hybrid Connection to connect back to on-premises resources from cloud.

Objective review

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of this chapter.

1. What VPN types are supported by Azure?
   A. Point-to-Site
   B. Site-to-Site
   C. vNet-to-vNet
   D. Multi-set

2. What’s the maximum bandwidth provided by ExpressRoute?
   A. 80 Mbps
   B. 200 Mbps
   C. 1 Gbps
   D. 10 Gbps

Objective 1.5: Describe Azure Services

Because your solution spans across multiple regions and facilities, you need to take additional care to ensure that the system performs at a global level. This objective introduces a couple of Azure services that can help you to optimize performance of a globally distributed system. Chapter 4 introduces more Azure services in the contexts of different scenarios.
Using Azure Traffic Manager

Traffic Manager routes incoming traffic to your application deployments at different geographic locations based on performance and availability.

To use Traffic Manager, you define a Traffic Manager profile that consists of a domain name, a list of endpoints, and a load-balancing policy. When a user tries to access a service, the following activities happen:

1. The user accesses the service by the domain name provided by Traffic Manager (*.trafficmanager.net). If a custom domain is used, another DNS resolution is performed to first resolve the custom domain name to the Traffic Manager domain name.
2. When Traffic Manager receives the DNS resolution request, it evaluates its policy and picks an endpoint address based on availability, performance, or a round-robin policy.
3. Traffic Manager returns a CNAME record that maps the Traffic Manager domain name to the selected endpoint.
4. The user’s DNS server resolves the endpoint address to its IP address and sends it to the user.
5. The user calls the endpoint directly by the IP address.

A couple of points are worth discussing here. First, Traffic Manager functions during the DNS resolution phase. The actual traffic doesn’t go through Traffic Manager. Second, because DNS records are often cached, Traffic Manager isn’t involved in every service request. Third, the endpoints don’t need to be on Azure. They can be on other cloud platforms, or even in on-premises datacenters.

Traffic Manager picks endpoints based on one of the following three methods:

- **Round-robin** Traffic is distributed to all endpoints evenly or based on weights.
- **Performance** Traffic Manager periodically updates a table that records the response time between various IP ranges to Azure datacenters. When a new request comes in, it picks the datacenter with the best response time in corresponding IP range.
- **Failover** Traffic Manager returns the primary endpoint by default. However, if the primary endpoint becomes unavailable, it will return backup endpoints according to their assigned priorities.
NOTE  DNS TIME-TO-LIVE (TTL)

DNS records can be cached on both DNS clients and DNS servers. The client will continue to use cached records until the record’s TTL expires. So, failing-over with Traffic Manager is not non-interruptive. Some customers will still be routed to the broken instances until the TTL expires.

These three methods are suitable for different scenarios. The round-robin method can be used for load-balancing in a same region or across multiple regions. The performance method can be used to optimize user traffic distribution. And the failover method can be used in failover scenarios.

You can also nest Traffic Manager profiles, which means a profile at a higher level uses other Traffic Manager endpoints as candidate endpoints. Using nested profiles, you can implement more complex policies. For example, you can have a top-level profile that uses the failover method to establish a primary site and a secondary site, and a second-level profile that distributes user traffics based on performance. You can have up to 10 levels of nested profiles.

Using CDN

Azure operates out of facilities located in 17 regions around the world, and that number is increasing every year. In addition, Azure also strategically places CDN point of presence (POP) locations to deliver content to end users. You can cache content from Azure Storage, Web Apps, and Azure Cloud Services.

When a user requests content by the CDN URL, the content is directly served from the CDN node, if the content exists. Otherwise, the content will be retrieved from the content origin and stored at the CDN node for future requests.

Using CDN has two major benefits. First, because content is served directly from the CDN node that is closest to the user, user experience can be greatly improved. Second, because a large portion of requests will be served from CDN nodes instead of from the original service nodes, the loads on the original service nodes are greatly reduced, making it possible for the service to scale-out to support a much greater number of users.

CDN is used mostly to cache static contents. However, you can cache dynamic outputs from your websites and cloud services as well because CDN content is identified by URLs, including the query parameters. For example, http://<identifier>.vo.msecnd.net/chart.aspx?item=1 and http://<identifier>.vo.msecnd.net/chart.aspx?item=2 represent two different cached objects. You need to be careful not to cache volatile data in CDN, because doing so can adversely affect your performance or even cause content problems, all at increased cost.
**Objective summary**

- Traffic Manager can distribute user traffic based on availability and performance.
- Traffic Manager uses the round-robin, performance, or failover method to decide to which endpoint to route traffic.
- CDNs serve cached content directly from CDN nodes that are closest to end users.
- CDNs can reduce traffic to original service nodes by serving static content directly.

**Objective review**

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of this chapter.

1. Which of the following are methods Traffic Manager uses to pick endpoints?
   - A. Round-robin
   - B. Failover
   - C. Performance
   - D. Random

2. What are the benefits of using a CDN?
   - A. Reduce response time
   - B. Reduce traffic to the original service
   - C. Improve data consistency
   - D. Enable faster upgrades
Answers

This section contains the solutions to the thought experiments and answers to the objective review questions in this chapter.

Objective 1.1: Thought experiment

1. There’s no single best way to explain how data is secured in the cloud. However, a simple analogy is quite effective: Ask if one would deposit money in a bank or keep cash under a couch cushion. Sure, the cash is closer to the owner when stored under the cushion, but the owner won’t be able to provide the level of protection a bank can offer. When you save data to Azure, your data is replicated at least three times for high availability. And Azure makes sure your data is accessible only by you.

2. Again, there’s no single correct answer. One possible approach is to talk about service recovery. Applications will fail, no matter where an application is deployed. The key to improving service availability is how quickly you can recover from errors. In traditional datacenters, MTTR is usually quite lengthy. Referring to previous service interruption cases if a good strategy to illustrate how reduced MTTR can help to dramatically increase service availability.

Objective 1.1: Review

1. **Correct answers:** A, B, C, and D
   
   **A.** **Correct:** Sufficient training is the foundation of building up a high-quality team.
   
   **B.** **Correct:** Automation is one of the most effective means to reduce human errors.
   
   **C.** **Correct:** Just-in-time access ensures that there’s no standing access to Azure resources, reducing the risk of accidental operations being carried out on customer data.
   
   **D.** **Correct:** Operation policies must be reinforced to ensure established workflows and practices are precisely followed.

2. **Correct answers:** A, B, C, and D
   
   **A.** **Correct:** Azure is committed to annual certification against ISO/IEC 27001/27002:2013.
   
   **B.** **Correct:** Azure has been granted a Provisional Authority to Operate (P-ATO) from the Federate Risk and Authorization Management Program (FedRAMP).
   
   **C.** **Correct:** Microsoft currently offers HIPPA Business Associate Agreement (BAA) to customers who have an Enterprise Agreement (EA).
   
   **D.** **Correct:** Microsoft offers customers European Union Standard Contractual Clauses.
3. **Correct answers:** B
   - **Incorrect:** Single-instance VMs don’t qualify for SLA.
   - **Correct:** Azure SLA requires at least two multi-instance VMs be deployed in the same Availability Set.
   - **Incorrect:** If an Availability Set only contains a single VM, the VM doesn’t qualify for SLA.
   - **Incorrect:** Two VMs must be in the same Availability Set to qualify for SLA.

**Objective 1.2: Thought experiment**

1. Although you can use both ACL and NSG to control network traffic to VMs, NSG is a better choice in this case because, 1) you can define rules that apply to a subnet instead of a VM, and 2) you can gain greater control by defining inbound rules and outbound rules independently.

2. One possible way to design the topology is to put Internet-facing resources, application servers, and database servers into different subnets. The Internet-facing resources can communicate only to application servers through specific ports. And only application servers can access database servers governed by another set of rules.

**Objective 1.2: Review**

1. **Correct answers:** A, B, C, and D
   - **Correct:** Each VM has an associated public virtual IP (VIP).
   - **Correct:** Each VM has one or multiple private IP addresses, one per NIC.
   - **Correct:** A static public IP can be associated with a VM.
   - **Correct:** A private static IP address can be associated to a VM on a virtual network.

2. **Correct answers:** A, B, and C
   - **Correct:** VIRTUAL_NETWORK denotes all IP ranges in the same virtual network, including connected networks.
   - **Correct:** AZURE_LOADBALANCER denotes the IP address of the Azure load balancer.
   - **Correct:** INTERNET denotes all IP addresses outside the virtual network.
   - **Incorrect:** VIRTUAL_MACHINE is not a default tag.
3. Correct answers: A, B, C, and D
   A. Correct: An NSG rule defines traffic flow control from a source range to a destination range. The source range is defined by source IP and source port.
   B. Correct: An NSG rule defines traffic flow control from a source range to a destination range. The destination range is defined by target IP and source port.
   C. Correct: You can apply an NSG rule to TCP, UPD, or * for both protocols
   D. Correct: Each NSG rule has an associated priority. Rules with lower priority can be overridden by rules with higher priorities.

4. Correct answers: A, B, C, and D
   A. Correct: Each ACL rule has a rule number, which denotes the priority of the rule.
   B. Correct: The remote subnet defines the IP range that the rule will be applied to.
   C. Correct: An ACL rule is associated with a VM endpoint.
   D. Correct: An ACL rule can be either a permitting rule or denying rule.

Objective 1.3: Thought experiment
1. Reliability, availability, security, and performance are all valid concerns. Especially, because Azure provides SLAs only if there are at least two VMs in an Availability Set, to ensure availability, you’ll need to deploy the application to at least two VMs and join them behind a load balancer. This might immediately cause some problems because not all applications are designed for such deployment. For instance, some of the legacy systems are designed to have a single central server that handles all user transactions. When the transactions are distributed to multiple instances, you might have two centers of truth that can’t be reconciled. Data replication and customer partition are two effective approaches in some cases.

2. To take full advantage of the cloud, you should explore the possibility of moving the application to PaaS. With VMs, you are still responsible for managing the virtualized infrastructure. With PaaS, you can focus almost entirely on implementing your business logics and leave the rest to Azure.

Objective 1.3: Review
1. Correct answer: C
   A. Incorrect: A-series is designed for generic workload, with A8 through A11 designed for HPC.
   B. Incorrect: D-series is designed for applications with high CPU and high temporary data IO.
   C. Correct: DS-series is designed for applications with high persisted data IO.
   D. Incorrect: G-series is for application with high CPU and memory demands.
2. **Correct answer:** D
   A. **Incorrect:** 8 is below limitations of any series.
   B. **Incorrect:** 16 is the limit of A-series.
   C. **Incorrect:** 32 is the limit of D-series and DS-series.
   D. **Correct:** G-series supports up to 64 data drives.

3. **Correct answer:** A
   A. **Correct:** Azure Resource Template uses JSON format.
   B. **Incorrect:** Azure Resource Template doesn’t support XML format.
   C. **Incorrect:** Azure Resource Template doesn’t support YAML format.
   D. **Incorrect:** Azure PowerShell is a scripting language, it’s not used to describe an Azure Resource Template.

4. **Correct answers:** A, B, C, and D
   A. **Correct:** Custom Script Extension downloads and runs configuration scripts such as DSC to designated VMs.
   B. **Correct:** Chef and Puppet are both integrated third-party solutions.
   C. **Correct:** Azure Automation can periodically check and fix your resource states so they don’t drift away from standard settings.
   D. **Correct:** Containerization is an effective way to pack applications as consistently deployable unit.

**Objective 1.4: Thought experiment**

1. Common techniques include introducing cache to reduce accesses to databases, using asynchronous IO operations, compressing data, sending deltas and only required data instead of complete data sets, and paging.

2. You can use queues to decouple components to break hard dependencies among services so that they can run at different paces. You can also consider SOA and Microservices to decompose complex applications into smaller services that can evolve separately.

**Objective 1.4: Review**

1. **Correct answers:** A, B, C, and D
   A. **Correct:** Use Point-to-Site connections to connect local compute to Azure Virtual Networks.
   B. **Correct:** Use Site-to-Site connections to connect on-premises network to Azure Virtual Networks.
   C. **Correct:** Use vNet-to-vNet VPN to connect two Azure virtual networks.
   D. **Correct:** Use Multi-site VPN to connect multiple on-premises networks to the same Azure virtual network.
2. **Correct answers:** D
   
   A. **Incorrect:** 80 Mbps is roughly the bandwidth a standard Azure Virtual Network gateway provides.
   
   B. **Incorrect:** 200 Mbps is roughly the bandwidth a high-performance Azure Virtual Network gateway provides.
   
   C. **Incorrect:** 1 Gbps is the maximum ExpressRoute bandwidth when a network service provider is used.
   
   D. **Correct:** 10 Gbps is the maximum ExpressRoute bandwidth when an exchange provider is used.

**Objective 1.5: Thought experiment**

1. In this case, the Traffic Manager policy will use the failover method, with a primary endpoint pointing to on-premises deployment and a secondary endpoint pointing to cloud deployment.

2. As the maintenance begins, the on-premises site is brought down. Some customers will still be redirected to the on-premises endpoint, leading to service interruption. As DNS records expires, new customer requests will be redirected to the cloud endpoint. You should note that this is not a zero-downtime solution.

**Objective 1.5: Review**

1. **Correct answers:** A, B, and C
   
   A. **Correct:** Traffic Manager supports the round-robin method that distributes traffic evenly to endpoints.
   
   B. **Correct:** Traffic Manager supports the failover method that routes traffic to the primary endpoint and then to the secondary endpoint when the primary is unavailable.
   
   C. **Correct:** Traffic Manager supports performance-based routing that picks the endpoint with the least response time.
   
   D. **Incorrect:** Traffic Manager doesn’t support random routing.
2. **Correct answers:** A and B

   A. **Correct:** CDN reduces response time by serving content directly from CDN locations.

   B. **Correct:** With static contents served from CDN locations, the traffic to the original service nodes can be greatly reduced.

   C. **Incorrect:** With CDNs serving cached contents, data could be out-of-sync with server versions and will eventually become consistent with server when local cache expires.

   D. **Incorrect:** CDN has nothing to do with your application server upgrades. On the other hand, because older static contents are served from CDNs, it will take time for the new static content to propagate to all CDN nodes.
Index

A

A8-A11 VMs, compute-intensive instances, 191
AAD Sync, 78
directory synchronization with, 79–82
access control, 98–102
Azure AD, 100
Azure SQL Database, 99
Azure Storage, 98
designing a role-based strategy, 109–121
access control challenges of large enterprises, 109
empowering users with self-service, 112
implementing RBAC, 110
improving security policies over time, 117
using Azure AD Access Panel, 115–116
using Azure AD Device Registration Service, 116–117
using RBAC for Azure resources, 111
for Azure customers’ data, 5
in other Azure services, 101
role-based access control
for resources in Resource Group, 240
access control lists (ACLs), 101
network, 20
Network Security Groups (NSGs) versus, 101
NSGs versus, 21
Access Control Service (ACS), 71, 87, 165
key concepts, 89
using with AD FS, 89–90
Access Panel, 70, 84
using, 115–116
Active Directory (AD)
Azure AD versus Active Directory Domain Services, 70
Federation Service (AD FS), 71
directory synchronization with, 77–82
using with Azure ACS, 89
Federation Services (AD FS), 165
on-premises, working with Azure AD, 70
Rights Management Services, 106
Rights Management Services (AD RMS), 97
Actor Pattern (or Actor Model), 225
ADAL (Azure AD Authentication Library), 69
sample scenario with Visual Studio, 71–73
AD DS (Active Directory Domain Services)
on-premises, versus Azure AD, 70
AD FS. See Active Directory (AD)
administrator roles (Azure AD), 100
Advanced Message Queuing Protocol (AMQP), 234
AES Clear Key dynamic encryption, 176
AES encryption, 239
Affinity Groups, 4, 14
alerts, 309
alternative services (for availability), 206
AMQP (Advanced Message Queuing Protocol), 235
analysis, 190, 234
Azure Stream Analytics, 235
Android, 141, 143
implementing Azure Mobile Services, 144
API Management, 51. See Azure API Management
App Controller, 308
monitoring capabilities, 317
volume and performance concerns, 311
AppDynamics, 209, 323
App Hosting Plan. See App Service Plan
Apple Push Notification, 153
Apple Push Notification Service (APNS), 237
application design, 189–250. See also web applications
creating compute-intensive applications, 190–202
application design

application design, continued
implementing Competing Consumers pattern, 200–201
understanding Azure Batch Apps, 199–200
using Azure Batch, 193–199
using Azure in high-performance environment, 190–193
creating long-running applications, 203–221
Cloud Service basics (sample scenario), 217–219
designing available applications, 203–207
designing reliable applications, 207–210
designing scalable applications, 211–213
using Azure Autoscale, 213–215
using Cloud Services, 215–217
VM Availability Set and autoscaling, 214–215
integrating Azure services in a solution, 229–242
building enterprise mobile applications, 236–238
creating data-centric web applications, 230–233
creating media applications, 238–239
managing related services, 240–241
working Big Data and Internet of Things, 233–236
selecting appropriate storage option, 221–229
Application Image (Batch Apps), 200
Application Insights, 208
integration with Cloud Services, 217
use case for, 324
using to debug websites, 258
using to monitor an ASP.NET application (sample scenario), 209–210
Application Proxy, 237
Application Proxy (Azure AD)
configuring, 82–85
applications
restarting after VM recovery, 10
scaling on virtual machines (VMs), 40–43
App Service BizTalk API Apps
Hybrid Connections, 170–171, 237
App Service BizTalk API Apps Hybrid Connection, 51
App Service Plan, 273–275
App Service Web Apps, 251–304
Azure virtual machines and Cloud Services, 263–267
comparison of service characteristics, 266
Web Apps, 265
creating websites using Visual Studio, 254–256
debugging websites, 257–259
deploying websites, 268–286
App Service Plan, 273–274
creating deployment packages, 271–273
deployment slots, 275–277
publishing options, 278–285
resource groups, 277–278
using Site Extensions, 269–271
designing websites for business continuity, 286–299
backing up and restoring data, 291–293
configuring data replication patterns, 289–291
designing for disaster recovery, 294
designing the data tier, 296–298
scaling with Web Apps and SQL Database, 287–289
updating websites with minimal downtime, 291
globally scaling websites, 252–253
monitoring of, 322
scaling websites, 161–163
supported languages, 259–263
Windows PowerShell commands for, 360
A records, 17
A-series VMs, 26
different names in different pricing tiers, 28
ASP.NET
creating and securing an application, 72
creating new Web Application for Mobile Services, 147
monitoring application using Azure Application Insights, 209–210
using identity providers with applications, 90–93
Web API, 158
implementing custom Web API, 159–160
Web Application template, 254
asynchronous operations, 332
attacks
Man-in-the-Middle attacks, 176
attacks, Azure measures against, 6
auditing
Azure AD features for, 118
authentication
Azure ACS as authentication broker, 87, 89
Azure AD, 69
basic workflow in claims-based architecture, 66
defined, 65
enterprise mobile applications, 236
external services, using with ASP.NET, 92
for Mobile Services, 149
for Web API applications, 165
multifactor, 111
multitiered applications in claims-based architecture, 68
native clients, in claims-based architecture, 67
OWIN and authentication middleware, 90
strategies used by Azure services, 11
System Center, using to connect to Azure, 310
Windows PowerShell to Azure, 354
authorization
Azure AD, 69
basic workflow in claims-based architecture, 66
defined, 65
mobile services, 149
multitiered applications in claims-based architecture, 68
native clients, in claims-based architecture, 67
automation
designing Azure automation and PowerShell workflows, 354–365
creating Windows PowerShell script for Azure, 354–363
use cases for Azure Automation configuration, 365–371
Azure Automation, 369
Chef and Puppet, 368
Desired State Configuration, 366–367
Windows PowerShell for automation, 368
automation options, 6
Automation service, 35, 240
autonomy of homogeneous instances, 205
Autoscale, using, 213–215
autoscaling, 41–43, 211
   App Service Web Apps, 287
   for websites, 161
   sample policy, 42
   using Azure Autoscale, 213–215
availability, 305
   Azure Apps or App Service Web Apps, 327
   business continuity and disaster recovery (BC/DR) plan and, 331
   Cloud Services features for, 216
deploying websites to multiple regions, 294–296
designing available applications, 203–207
   homogeneous instances, 205
   primary-secondary instances, 205
   Service Level Agreements for some Azure services, 204
   single-component availability, 204
   system-level availability, 206
effects of scalability, 335
high availability and Azure SLAs, 334
of data storage, 226
of infrastructure, 7
updating Web Apps with minimal downtime, 291
VM configurations qualifying for SLA, 8
Availability Sets, 8, 42, 327, 328, 334
autoscaling rules, 214
AzCopy, 135
Azure
   global reach of, 2
   regions and datacenters, 3
   managing Azure resources with System Center, design considerations, 310–312
   System Center components hosted in, 306–309
   Azure Active Directory (Azure AD), 69
   access control, 100
   Access Panel, 115
   Active Directory Domain Services (AD DS) versus, 70
   ADAL and Visual Studio, sample scenario, 71–73
   creating and securing ASP.NET application, 72–73
   provisioning Azure AD tenant and user, 71
   auditing and reporting feature to monitor security activities, 118
   authentication for enterprise mobile applications, 236
   authentication for Web API applications, 165
   Cloud App Discovery service, 117
   data reliability, 104
   Device Registration Service, 116–117
   Free, Basic, and Premium tiers, 70
   Graph API, 74–75
   structure of queries, 74
   using the client library, 75
   module for Windows PowerShell, 79
   reacting to problems or security infractions, 119
   securing resources using hybrid identities, 77–86
   configuring Azure AD Application Proxy, 82–85
Azure Active Directory (Azure AD), continued
- support by Mobile Services, 142
- working with on-premises AD, 70
- Azure AD Authentication Library (ADAL), 69
- AZURE_LOADBALANCER tag, 21
- Azure Management Pack for Operations Manager, 317
- Azure Marketplace
  - prebuilt web apps available on, 266
- Azure Trust Center, 96
- Azure Virtual Machine Agent. See VM Agent
- AzureWatch, 323
- Azure WebJobs, 163–165
- Azure Websites. See also App Service Web Apps
  - name change, 251
  - scaling up to increase performance and throughput, 161
  - support for BizTalk API Hybrid Connections, 170
- Windows PowerShell commands for, 360

B
- backups
  - characteristics of properly designed backup solutions, 342
  - Data Protection Manager, 307
  - offsite backup and a hybrid scenario, 313
  - using Azure Backup, 343–347
  - using Data Protection Manager, 347–349
  - using StorSimple, 350–351
- Backup service, 104, 341, 343–347
  - deploying and configuring, 343–346
  - recovering data, 346
  - using Azure PowerShell to perform tasks, 346
- backup window, 311
- BACPAC files (SQL Database), 103
- bandwidth
  - concerns in designing hybrid management solution, 311
- Batch account, 193
- Batch API, 193
- Batch Apps API, 193
- Batch service, 193–199, 243
  - face detection with Batch (sample scenario), 194–200
  - overview, 193
  - understanding Batch Apps, 199–200
  - preparing a Batch Application, 200
- Billing administrator role (Azure AD), 100
- Binary Large Objects (Blobs) storage, 130
  - application logging stored in, 323
  - saving files, 132
  - security options, 137
  - uploading local file to Storage account, 131
- BitLocker encryption, 97, 136
- BizTalk API Apps. See App Service BizTalk API Apps
- bring your own devices (BYOD), 110
- business continuity, designing websites for, 286–299
  - backing up and restoring data, 291–293
  - configuring data replication patterns, 289–291
  - deploying websites to multiple regions for high availability, 294
  - designing for disaster recovery, 294
  - designing the data tier, 296–298
  - scaling with App Service Web Apps and SQL Database, 287–289
  - updating websites with minimal downtime, 291
- business continuity/disaster recovery (BC/DR) capabilities, designing, 330–341
  - architectural capabilities of, 331–336
  - Hyper-V Replica and Azure Site Recovery, 336–338
  - use cases for Hyper-V Replica and Azure Site Recovery, 338–340

C
- C#, 259, 263
  - developing for Azure Mobile Services, 143
- caching, 230
  - service throttling and, 10
  - storing repetitive query results, 224
- Capacity Units (DocumentDB), 134
- CDNs (content delivery networks), 54
- CDN service. See Content Delivery Network service
- CENC (common encryption scheme), 177
- certificates, 310
  - authenticating Windows PowerShell to Azure, 354
- Changed Block Tracking, 347
- Chef, 35, 240, 365, 368
China, Azure in, 4
claim rule (Azure ACS), 89
claims, 64
claims-based architecture, 64
    basic authentication and authorization workflow, 66
    native clients, authentication and authorization process, 67
other scenarios, 69
RBAC and claims, 111
working with multitiered applications, 68
Classless Inter-Domain Routing (CIDR) notation, 15
ClearDB, 134
    website, database size and performance settings, 138
cloud
defined, 1
designing datacenters for, 8–11
Cloud App Discovery, 117
cloudapp.net domain, 216
Cloud Assembly (Batch Apps), 200
Cloud Distribution Point, 307
CloudNinja Metering Block, 323
Cloud Services, 215–217
    availability, reliability, and scalability of, 216
    comparison with App Service Web Apps and VMs, 266
    creating and implementing (sample scenario), 217–219
    creating websites, 263
    endpoints and load balancing, 216
    joining to a domain, 172
    monitoring and recovery of application process, 10
    roles and instances, 215
    setting up a new Cloud Service, 264
    virtual machines (VMs) and, 16
clusters
    constructing an HPC cluster, 191–193
        Azure Premium storage and large VM instances, 192
        configuring HPC cluster head node, 192
CNAME records, 17, 53
code, capturing infrastructure as, 36–39
Column-Level Encryption (CLE), SQL Server, 98
custom encryption scheme (CENC), 177
Competing Consumers pattern, implementing, 200–201, 217–219
compliance, 7
compute-intensive applications, creating, 190–202
    compute-intensive instances, 191
    constructing an HPC cluster, 191–193
    implementing Competing Consumers pattern, 200–201
    understanding Azure Batch Apps, 199–200
    using Azure Batch, 193–199
Compute (Preview Management Portal), 24–45
concurrency
    optimistic versus pessimistic, 222
Configuration Manager, 307
    services provided for Windows and Linux, 314
    use case for, 325
    using to manage hybrid infrastructure, 313
connection strings
    setting to be sticky in Preview Management Portal, 276
connector space, 78
consistency
    immediate versus eventual, 222
Consumer Groups, 234
containers, 38
    benefits of use for reactive scaling, 212
    Blob file storage in, 131
    decoupling compute and resource, 39
    higher compute density with, 39
    orchestration, 39
    security options for, 137
Content Delivery Network (CDN) service, 54, 232
    replicating data with, 289
content delivery networks (CDNs), 54, 230
    using to globally scale websites, 252
    website deployment geographically close to users, 295
Contributor role, 111
Cross-Platform Command-Line Interface (xplat-cli), 6, 119, 240
cross-region deployments, 206
CustomerEntity class, 131
Custom Script Extension, 34
data

D

data
- amounts collected and used, 129
- application design, working Big Data and Internet of Things, 233–236
  - Azure Event Hubs, 234
  - Azure Machine Learning, 235
  - Azure Stream Analytics, 235
- canonical Big Data pipeline components, 233
- map of Azure services to Big Data pipeline, 234
- sample architecture, 236
- Big Compute and Big Data, 190
- designing data tier for a website, 296–298
  - loss of, acceptable amount, 331

data access
- data query patterns, 223
  - repetitive queries, 224
  - static versus dynamic schema, 223
- designing a strategy for hybrid applications, 168–174
  - connecting to on-premises data using Service Bus Relay, 169
  - identifying constraints for connectivity via VPN, 172
  - options for domain-joining Azure VMs and Cloud Services, 172–173
  - Web Apps VPN capability, 171
  - read/write patterns, 222
  - immediate versus eventual consistency, 222
  - optimistic versus pessimistic concurrency, 222
  - sequential versus random access, 223
- database management system (DBMS), 308
- databases. See also MongoDB; MySQL; NoSQL; SQL Database
  - choosing type for data tier of website, 296
  - Database Throughput Units (DTUs), 137
- datacenters
  - Azure's use of Global Foundation Services (GFS)
  - datacenters, 2–12
- data collectors, 233
- data dependent routing (DDR), 226
- data disks, 29–30
- data ingestors, 233

data loss, 331
- data producers, 233
- Data Protection Manager, 307, 311, 342
  - backups with, 347–349
    - Azure Backup of Data Protection Manager data, 348
    - connecting Data Protection Manager to Azure, 347
    - deploying Data Protection Manager, 348
    - workloads and versions supported, 348
  - use cases for, 351
  - using with Azure to store offsite data, 313
- data reliability and disaster recovery services, 102–108
  - Azure AD, 104
  - Azure Backup, 104
  - Azure Rights Management Services (RMS), 106
  - Azure SQL Database, 103
  - Azure Storage, 102
  - security key management with Azure Key Vault, 107
  - StorSimple, 105
- data replication. See replication
- data security
  - identifying an appropriate data security solution, 95–108
  - data protection technologies, 96–98
  - implementing effective access control policies, 98–102
- data sharding, 226, 335
- data storage, 129–140. See also Storage service
  - designing security options for SQL Database or Storage, 136–137
  - designing storage options, 130–136
    - DocumentDB, 134
    - MongoDB, 133–134
    - MySQL, 134
    - Storage account, 130–132
    - tools and libraries, 135
  - identifying appropriate VM type and size for solutions, 137–139
  - selecting appropriate storage option, 221–229
  - cost matters, 225
  - evaluating qualities of data storage, 226–228
  - keeping data close to compute, 225
  - understanding data access patterns, 222–224
  - using combination of stores, 224
Data Transfer Object (DTO), 141

data transformation, 234

DDR (data dependent routing), 226
deployment
cross-region deployments, 206
deploying websites, 268–286
App Service Plan, 273–274
creating deployment packages, 271–273
deployment slots, 275–277
implementing Azure Site Extensions, 269–271
publishing options, 278–285
resource groups, 277–278
deploying websites to multiple regions for high availability, 294
HPC clusters, using a deployment script, 192
multiregion deployment of Cloud Services with Traffic Manager, 217
swap deployment in Cloud Services, 216
System Center, 309–310
deployment slots
specifying for restored Web Apps files, 293
using to minimize website downtime, 291
designing advanced applications. See application design
Desired State Configuration, 34, 365, 366–367
in Azure, 369
practical applications of, 366
Device Registration Service (Azure AD), 116–117
DevOps movement, 36
diagnostics
built-in, in Cloud Services, 217
Diagnostic Service, 317
Diagnostics service, 208
use with Cloud Services, 217
Digital Rights Management (DRM), 176
media application content protection, 239
DIP (dynamic IP) addresses, 16
directory synchronization, 78
setting up, sample scenario, 79–82
DirSync, 78
disaster, 332
disaster recovery
designing App Service Web Apps for, 294
designing a strategy for, 341–353
backup solutions for Azure, 342–351
use cases for StorSimple and Data Protection Manager, 351–352
designing for Azure, 330–341
using Azure in a hybrid configuration, 313
disaster recovery services. See also data reliability and disaster recovery services
Azure Site Recovery, 105
D language, 265
DNS (Domain Name System)
Azure Traffic Manager, 53
eventual consistency of records, 222
name resolution and DNS servers, 17
Docker, 212
DocumentDB, 134, 231, 296
DOM Explorer, 258
DRM. See Digital Rights Management
Dropbox, 282–285
DSC. See Desired State Configuration
D-series and DS-series VMs, 193
D-series VMs, 27
DTO (Data Transfer Object), 141
DTUs (Database Throughput Units), 137
dynamic enterprise environment, security policies in, 110
dynamic IP (DIP) addresses, 16
Dynamic Packaging, 176
dynamic scaling, 211
dynamic schemas, 223

E

Elastic Scale (SQL Database), 226–227
Emgu 2.9, downloading and installing, 198
Encrypting File System (EFS), 97
encryption, 7
Azure RMS, 106
for data in Azure Storage, 97
managing security keys with Azure Key Vault, 107
SQL Server Column-Level Encryption (CLE), 98
SQL Server Transparent Data Encryption (TDE), 97
SSL, use by SQL Database, 136
supported by Media Services, 176
Endpoint Protection, 309
endpoints

endpoints
Cloud Services, 216
methods of choosing by Azure Traffic Manager, 53
virtual machines (VMs) on Azure, 18
end-user capabilities with backup solutions, 342
errors
in cloud datacenters, 5
transient, 207
Event Hubs, 234
eventual consistency, 222
ExpressRoute, 45, 48–49, 314
connecting resources in Azure to corporate network, 171
external state store, 205

G
Geo Redundant Storage (GRS), 332
Geo-Redundant Storage (GRS), 103, 130
geo-replication, 290, 332
GFS. See Global Foundation Services datacenters
GIT
using for website backups, 291
website deployment with, 279–282
Global administrator role (Azure AD), 100
Global Foundation Services (GFS) datacenters, Azure
use of, 2–12
Azure's global footprints, 2–4
regional differences, 4
regions and datacenters, 3
designing cloud-scale datacenters, 4–8
errors, 5
human factors, 5
sustainable reliability, 7
trust-worthy computing, 6
designing for the cloud, 8–11
datacenter maintenance, 8–10
datacenter outages, 10
service security, 11
service throttling, 10
Global Service Manager
use case for, 323
Google
support by Azure ACS, 90
face detection with Batch (sample scenario), 194–200
failover
automatic, for critical applications, in Traffic Manager, 206
failover option in load balancing, 53, 295
failures in cloud datacenters, 5
Fault Domains, 10, 207, 327, 328
fault tolerance, 330
Federation Migration Utility, 227
files
in Azure Batch, 194
Files storage, 130, 132
application logging stored in, 322
File Storage API, 132
FIM + Azure AD Connector, 78
FQDNs (fully qualified domain names), 17
Fragmented MP4, 177
FTP
deploying WebJobs over, 165
using to publish App Service Web Apps, 279
fully qualified domain names (FQDNs), 17

F
Facebook, 129
support by Azure ACS, 90
face detection with Batch (sample scenario), 194–200
failover
automatic, for critical applications, in Traffic Manager, 206
failover option in load balancing, 53, 295
failures in cloud datacenters, 5
Fault Domains, 10, 207, 327, 328
fault tolerance, 330
Federation Migration Utility, 227
files
in Azure Batch, 194
Files storage, 130, 132
application logging stored in, 322
File Storage API, 132
FIM + Azure AD Connector, 78
FQDNs (fully qualified domain names), 17
Fragmented MP4, 177
FTP
deploying WebJobs over, 165
using to publish App Service Web Apps, 279
fully qualified domain names (FQDNs), 17

H
Hadoop, 233
Hardware Security Module (HSM), 107
health monitoring, 208
high-performance computing (HPC) environment, using Azure in, 190–193
Azure Premium storage and large VM instances, 192
compute-intensive instances, 191
constructing an HPC cluster, 191
homogeneous instances, 205
host names, 17
HPC Pack, 191
HSM (Hardware Security Module), 107
HTML, 141
HTML/JS, 143
  implementing Azure Mobile Services, 144
HTTP
  Hybrid Connections, 170
HTTP headers
  Azure AD Graph API requests, 74
human factors in cloud datacenters, 5
Hybrid Connections, 170, 237
  configuring, 171
hybrid identities. See identities, hybrid
hybrid scenario, circumstances that dictate, 312
Hyper-V Replica, 336–338
  and Azure Site Recovery, use cases for, 338–340

IaaS (Infrastructure as a Service), 1
  MongoDB in Azure Linux VMs, 133
  SQL Server running in a VM, 133
identities, hybrid, 77–86
  configuring Azure AD Application Proxy, 82–85
  setting up directory synchronization with AD FS, 77–82
identities, managed, 63. See also Azure Active Directory
  identity providers, 64
  deciding whether to use an external provider, 75
for Mobile Services, 148
  in Azure ACS, 89
  using to secure resources, 86–95
  Azure ACS with social networks, 90
  external identity providers with Azure Mobile Services, 94
  identity providers with ASP.NET applications, 90–93
images
  capturing custom images, 31
  OS, 30
  sources for Azure VM, 31
  using custom images, 32
  VM, 25
  immediate consistency, 222
Import and Export Service (SQL Database), 103
Import/Export service, 135
inbound rules of an NSG, 21
indexing
  advanced techniques in NoSQL databases, 224
  Media Indexer, 177
Information Technology Infrastructure Library (ITIL), 308
infrastructure and networking, designing, 1–62
Azure Compute, 23–45
  capturing infrastructure as code, 36–39
  managing images, 31–32
  managing VM states, 33–36
  scaling applications on VMs, 40–43
  selecting VM sizes, 24–30
Azure virtual networks and networking services, 12–23
  access control lists (ACLs) and network security groups (NSGs), 18–22
  creating cloud-only network, 13–18
Azure virtual private network (VPN) and ExpressRoute, 45–52
  describing Azure services, 52–55
  using CDN, 54
  using Traffic Manager, 53
how Azure uses Global Foundation Services (GFS) datacenters, 2–12
  Azure’s global footprints, 2–4
  designing cloud-scale datacenters, 4–8
  designing for the cloud, 8–11
Infrastructure as a Service. See IaaS
Input Endpoints, 216
Instance Input Endpoints, 216
Instance-Level Public IP (PIP) address, 17
Intel MPI Library, Azure support for, 191
Internal Endpoints, 216
Internal Load Balancers (ILBs), 40
Internet Information Services (IIS), 264
Internet of Things (IoT), 102
  application design incorporating, 233–236
Internet Protocol Security (IPSec), 310
  using with Operations Manager gateway, 319
libraries
for interactions with data storage systems and accounts, 135
lifecycle boundary, 240
Lightweight Directory Access Protocol (LDAP), 70
LINQ (Language Integrated Query), 75
Linux
  Configuration Manager services for, 314
  infrastructure in Azure, automating and managing, 368
Linux VMs
  creating a MongoDB in, 133
  running websites, 265
  live streaming, 177
  load-balanced endpoints, 19
  load-balanced sets, 10
  Load Balancer, 18. See Azure Load Balancer
  load balancing, 40, 207
    Cloud Services, 216
    network load balancer in front of VMs in Availability Set, 334
    on Traffic Manager, 295
Locally Redundant Storage (LRS), 103, 130
long-running applications, creating, 203–221
  Cloud Services basics (sample scenario), 217–219
  designing available applications, 203–207
  designing reliable applications, 207–210
  designing scalable applications, 211–213
  using Application Insights (sample scenario), 209–210
  using Azure Autoscale, 213–215
  using Cloud Services, 215–217
  virtual machine Availability Set and autoscaling, 214
long-term storage, 234
loose coupling of components, 208

Machine Learning service, 235
maintenance
  datacenter, 8–10
  managed identities. See identities, managed

INTERNET tag

INTERNET tag, 21
Intune, 307
iOS, 141, 143
  implementing Azure Mobile Services, 144
IP addresses
  for VMs in virtual networks, 16
  IP ranges in NSG rules, 20

Java, 260
JavaScript
  developing for Azure Mobile Services, 144
  MobileService Client implementation, 143
  using Node.js for Web Apps, 260
  using Node.js to create mobile services, 147
jobs, 194
JSON
  Azure Resource Templates, 37
  DocumentDB NoSQL database for documents, 231
  JSON document-based systems, 134
  raw JSON for MobileServiceClient, 143
JSON Web Token (JWT) format, 176

Kafka, 233
Katana (OWIN implementation), 90
Kerberos, 70
key management
  in Azure Storage, 97
  in SQL Server, 98
  using Azure Key Vault, 107
Key Vault, 97, 107

Language Integrated Query (LINQ), 75
last-write wins, 223
layered storage, 225
LDAP (Lightweight Directory Access Protocol), 70

Machine Learning service, 235
maintenance
  datacenter, 8–10
  managed identities. See identities, managed
management

design considerations, managing Azure resources with System Center, 310–312
simple, with backup solutions, 342
using Configuration Manager to manage hybrid infrastructure, 313
Management API, 6, 240
Management Packs, 309
Management Portal
autoscaling with, 213–215
creating a virtual network, 13–16
Man-in-the-Middle attacks, 176
McAfee Endpoint Security extension (VMs), 33
MCIO (Microsoft Cloud Infrastructure and Operations), 2. See also Global Foundation Services (GFS) datacenters
mean time between failures (MTBF), 5
mean time to recover (MTTR), 5
Media Indexer, 177
Media Services, 175–179, 238
key components, 176–178
overview, 175–176
media solutions, designing, 175–179
Azure Media Services overview, 175–176
creating media applications, 238–239
ingredients, 238
sample architecture, 239
key components of Media Services, 176–178
messaging system (Queue storage), 132
metaverse, 78
MFA. See multifactor authentication
Microsoft account
support by Azure ACS, 90
Microsoft Azure Trust Center, 7
Microsoft Cloud Infrastructure and Operations (MCIO), 2. See also Global Foundation Services (GFS) datacenters
Microsoft HPC Pack, 191
Microsoft Operations Framework (MOF), 308
Microsoft Rights Management for Individuals, 106
Microsoft Rights Management Service (or Azure RMS), 106
Microsoft System Center. See System Center
Microsoft Update, 326
Microsoft Visual Studio. See Visual Studio
Migration Accelerator, 338

mobile applications
building enterprise mobile applications, 236–238
Azure App Service BizTalk API Apps Hybrid Connections, 237
Azure Notification Hubs, 237
map of Azure services and mobile applications, 237
sample architecture, 238
using external identity providers with Azure Mobile Services, 94
MobileServiceClient object, 143, 150
Mobile Services, 141–153
consuming Mobile Services, 143–145
extending using custom code, 150–151
hosting options, 142
implementing, 147–148
implementing push notification services, 153–155
Offline Sync, 145–146
security, 148–149
authorization, 149
using external identity providers with, 94
Mobile Services Client Library, 143
modeling, 190
Model-View-Controller (MVC) pattern
for Web API application, 159
MongoDB, 133, 226
adding to your Azure account, 134
cross-region replica set, 49
performance and size settings, 138
MongoLab, setting up an account on, 133
monitoring
data and Media Services, 177
designing a monitoring strategy, 316–330
Azure architecture constructs, effects on patching strategy, 327–329
built-in Azure capabilities, 322–323
Microsoft products and services for monitoring
Azure solutions, 317
System Center capabilities for monitoring Azure solutions, 317–321
third-party and open-source monitoring tools, 323
use cases for Operations Manager, Global Service Monitor, and Application Insights, 323–324
monitoring, continued
use cases for WSUS, Configuration Manager, and custom solutions, 325–326
simple, with backup solutions, 342
using Monitor component for web applications, 257
using Operations Manager, 308
MTBF (mean time between failures), 5
MTTR (mean time to recover), 5
multifactor authentication (MFA), 65, 111
Multi-Factor Authentication service, 111
multi-instance VMs, 8
multishard query (MSQ), 227
Multi-Site VPN, 50
multitenancy, 227
multitiered applications
authentication/authorization in claims-based architecture, 68
MVC (Model-View-Controller) pattern for Web API application, 159
MySQL, 134, 226
installing on an Azure VM, 135
performance and size settings, 138

N
namespaces (Azure ACS), 89
native clients
authentication and authorization process, 67
.NET, 259
using to create mobile services, 147
.Net Storage Client Library, 132
Network Access Protection, 314, 325
network interfaces (NICs)
creating for virtual machines on virtual networks, 17
Network Security Groups (NSGs), 101
access control lists (ACLs) versus, 21
ACLs versus, 101
creating, 20
New Relic, 209, 323
Node.js, 260
Azure Marketplace options, 261
creating mobile services, 147
NoSQL, 221, 296
advanced indexing techniques, 224

DocumentDB, 134
schema-less databases, 223
Table storage, 130
Notification Hubs, 237
notifications
designing applications that use, 153–158
push notification services in Mobile Services, 153–155
sending push notifications, 155–157
Mobile Services, push notification system, 142
NSGs (Network Security Groups), 101

O
OAuth 2.0 On-Behalf-Of (draft) specification, 69
Objective-C, 144
Offline Sync (Mobile Services), 145
offsite storage
scalable and secure, with backup solutions, 342
Open Data (OData) protocol, 74, 131
OpenID Connect, 70
Open Web Interface for .NET (OWIN), 90
and authentication middleware, 90
cookie authentication with ASP.NET, sample scenario, 92
operating system (OS) drives, 26
operating systems
running Azure VMs, 265
Operations Manager, 308, 316
monitoring Azure solutions, 317–321
direct connect proxy agent through VPN tunnel, 318
gateway in Azure, no VPN, 320
on-premises Operations Manager installation, 321
requirement for a hybrid scenario, 312
using with Global Service Monitor and Application Insights, 324
optimistic concurrency, 222
Orchestrator, 307
volume and performance concerns, 311
outages, datacenter, 10
outbound rules of an NSG, 21
OWIN. See Open Web Interface for .NET
Presentation and action, 234
ACL settings in, 101
app settings and connection strings, 276
Compute, 24–45
configuring autoscaling, 213
deployment slots, swapping, 275
pricing tiers, 26
primary-secondary instances, 205
privacy of customer data, 6
programming languages
support by Azure Web Apps, 252, 259
Java, 260
.NET, 259
Node.js, 260
PHP, 261
Python, 262
supported by client libraries for interaction with storage systems, 135
supported by VMs, 265
protocols
supported by Azure AD, 69
supported by NSG, 20
Public IP (PIP) address, 17
public-key infrastructure (PKI) certificates, 310
use in Operations Manager monitoring of Azure resources, 319
publishing
options for App Service Web Apps, 278–285
Dropbox, 282–285
FTP locations, 279–280
source control systems, 280–282
Puppet, 35, 240, 365
automating and enforcing consistency of systems, 368
push notifications, 153
Azure Notification Hubs, 237
implementing services in Mobile Services, 153–155
sending, 155–157
Python, 262
Azure Marketplace options, 263

Owner role, 111

P

PaaS (Platform as a Service), 1
Azure App Service Web Apps, 265
Azure Mobile Services, 141
Azure SQL Database, 97, 133
container-based offerings, 39
partitioning
in Event Hubs, 234
workload, 212
Password administrator role (Azure AD), 100
passwords
resets and registration, monitoring with Azure AD, 118
self-service resets, 113
patching
Azure architecture, effects on patching strategy, 327–329
custom patch management for applications, 326
performance levels for databases, 138
performance option in load balancing, 53, 295
pessimistic concurrency, 222
PHP, 261
Azure Marketplace options, 262
PIP (Public IP) address. See PaaS
Platform as a Service. See PaaS
Platform Notification Service (PNS), 153
Platform Notification Systems (PNS), 237
PlayReady DRM, 176, 239
PNS. See Platform Notification Service
Point-to-Site VPN, 46
PowerShell (Azure), 6, 101, 240
adding deployment slots to web applications, 277
creating new VM using a custom image, 32
downloading and editing Resource Templates, 37
Get-AzureVMAvailableExtension cmdlet, 33
OB (Online Backup) command, 347
performing tasks in Backup, 346
Workflows, 35
PowerShell Deployment Toolkit, 310
PowerShell, Windows. See Windows PowerShell
power usage effectiveness (PUE), 7

Python
QoS

QoS. See Quality of Service

Quality of Service (QoS), 189

Queue storage, 130, 132
  adding new message to an Azure Queue, 132

R

random access, 223

RBAC. See role-based access control

RCA (Root Cause Analysis) reports, 10

RDBMS. See relational databases

reactive scaling, 212

Read-Access Geo-Redundant Storage (RA-GRS), 103, 130

Reader role, 111

read/write data access patterns, 222

Real Time Messaging Protocol (RTMP), 177

recovering data
  in Azure Backup, 346
  in Data Protection Manager, 348

Recovery Point Objective (RPO), 331
  backup solutions meeting, 342
  low, meeting with StorSimple, 351

Recovery Time Objective (RTO), 331
  backup solutions meeting, 342
  low, meeting with StorSimple, 351

Redis, 226

redundancy, 330
  storage redundancy in Azure, 332
  using to improve service availability, 204

regions (Azure), 3
  cross-region deployments, 206
  cross-region MongoDB replica set, 49
  differences in, 4
  region-wide datacenter outages, 10

relational databases, 221, 224, 296
  lossless schema updates, 223

reliability
  Azure datacenters, 7
  Cloud Services features for, 216
  designing reliable applications, 207–210
  of data storage, 226

relying party, 64
  in Azure ACS, 89

Remote Direct Memory Access (RDMA), 191

repetitive queries, 224

replication
  configuring patterns for Web Apps, 289–291
  Hyper-V Replica, 336
  in Azure Storage, 226
  management by Site Recovery, 313

replication setting (Storage accounts), 130

reports
  produced by Azure AD, 118

Resource Groups, 25, 38, 112, 240
  scripting options to manage, 240
  using with Web Apps, 277–278

resources, 112
  decoupling from compute with containers, 39
  resource scopes, 111

Resource Templates, 37

RESTful API
  adding deployment slots to web applications, 277

RESTful Graph API, 70, 74–75

RESTful interfaces
  accessing Blobs in Storage, 131
  support by Azure AD, 70
  using to interact with storage systems, 135

restoring data
  backup solutions and, 342
  bandwidth considerations in hybrid management solution, 311
  in App Service Web Apps, 291, 293
  with StorSimple, 351

REST services
  implementing in Azure using ASP.NET Web API, 160

Rights Management Services (RMS), 97
  overview, 106

role-based access control (RBAC), 69, 109–121, 240
  access control challenges faced by large enterprises, 109
  empowering users with self-service, 112
  implementing RBAC, 110
    claims and, 111
    groups, 110
    multifactor authentication, 111
    roles, 110
improving security policies over time, 117
managing devices with Azure AD Device Registration Service, 116–117
using Azure AD Access Panel, 115
enabling SSO access to Twitter (sample scenario), 115
using RBAC for Azure resources, 111
roles and resource scopes, 111
using roles properly with SQL Database, 136
roles, 111
Cloud Services, 215
in web application (example), 233
inter-role communications in Cloud Services, 217
Root Cause Analysis (RCA) reports, 10
round-robin option in load balancing, 53, 295
Routing and Remote Access Service (RRAS), 47
RPO. See Recovery Point Objective
RRAS (Routing and Remote Access Service), 47
RTMP (Real Time Messaging Protocol), 177
RTO. See Recovery Time Objective
Ruby on Rails, 265
rule groups (Azure ACS), 89
runbooks, 35

S

SaaS (Software as a Service), 1, 69
access to applications, 109
discovering services used via Cloud App Discovery, 117
MongoLab servers, creating an account on, 133
SAML 2.0, 70
SAP (Shared Access Policy)
use with Azure Storage, 137
SAP (Stored Access Policy), 99
SAS. See Shared Access Signatures
scalability
effects on availability, 335
features of Cloud Services, 217
of data storage, 226
scaling
designing scalable applications, 211–213
dynamic scaling, 211
globally scaling websites in Web Apps, 252–253
in-depth information on, 336
scale-up and scale-out with Web Apps and Azure SQL Database, 287–289
system, 215
using Azure Autoscale, 213–215
virtual machine Availability Set and autoscaling, 214
website, using Azure App Service Web Apps, 161–163
scaling out, 211, 335
using workload partitioning, 212
scaling-out, 40
scaling up, 211, 335
scaling-up, 40
scheduled scaling, 212
schemas
static versus dynamic, 223
Scout, 338
scripting options to manage Azure Resource Groups, 240
scripts
automated scripts in containers, 39
using to manage VM states, 34
Search service, 231
securable entities, 64
Secure Socket Tunneling Protocol (SSTP), 46
Secure Sockets Layer, See SSL
Secure Token Systems (STS) providers, 176
securing resources
identifying an appropriate data security solution, 95–108
security
Azure datacenters, 6
creating secure notification system, 155
customers’ needs versus performance, 7
designing for Azure SQL Database or Storage, 136
for Mobile Services, 148–149
for streaming media, 176
for Web API, 165–167
monitoring, using Azure AD auditing and reporting, 118
of services, 11
System Center Endpoint Protection, 309
security context, 240
security keys, managing. See key management
security policies

security policies
    empowering users with self-service, 112
    improving over time, 117
    in a dynamic enterprise environment, 110
security tokens, 64
    sequential access, 223
Server Message Block (SMB) 2.1 protocol, 132
Service administrator role (Azure AD), 100
Service Bus
    access control, 102
Service Bus Relay, 51, 237
    connecting to on-premises data with, 169
service configuration file, 219
Service Configuration Schema, 219
service definition file, 219
Service Definition Schema, 219
service identities (Azure ACS), 89
Service Level Agreements (SLAs), 203
    for some Azure services, 204
    sample Azure service SLAs, 333
Service Manager, 308
service provider/relying party, 64
services
    alternative services (for availability), 206
    states, managing for external services, 35
service throttling, 10
session affinity, 205
session state management
    in use of homogeneous instances, 205
sharding, 226, 335
shard map management (SMM), 226
Shared Access Policy (SAP)
    use with Access Storage, 137
Shared Access Signatures (SAS), 11, 98
    generating using Azure Storage Client Library for .NET, 98
    on Service Bus, 102
    use with Azure Storage, 136
Simple Web Token (SWT) format, 176
simulation, 190
single-instance availability, 204
single-instance VMs, 9
Single Point of Failure (SPOF), 206
    centralized databases, 225
    fault domains, 327
Single Sign-On (SSO), 65
    enabling SSO access to Twitter, using Azure AD Access Panel, 115
Site Control Manager (SCM), 269
    adding new Site Extension on website, 270
    using to debug websites, 258
Site Extensions (App Service Web Apps), 269–271
    developing a new Site Extension, 270
Site Recovery, 105, 313, 336–338
    and Hyper-V Replica, use cases for, 338–340
    backups and restores of on premises VMs, 294
setup, 336
Site-to-Site VPN, 47, 314, 317
SLAs. See Service Level Agreements
SMB (Server Message Block) 2.1 protocol, 132
social networks
    as identity providers, using with Azure ACS, 90
    support by Mobile Services for providers, 142
    soft delete for data (Mobile Services), 142
Software as a Service. See SaaS
solid-state drives (SSDs), 27
    Premium storage and Azure compute-intensive VMs, 193
source control systems, 280–282
SQL Database, 97, 133, 296
    access control policy, 99
    connection to Mobile Services, 141
    data reliability, 103
    data replication and disaster recovery, 294
    designing security options for, 136
    Federation applications, migrating to Elastic Scale, 227
    geo-replication, configuring, 290
    performance levels, 138
    reliability of data storage, 226
    roles controlling access to, 111
    SQL Sync, 289
    tiers of service, 137
SQL Database Elastic Scale, 226–227, 288
SQL databases on Azure, 97
SQLite, use by Mobile Services Offline Sync, 145
SQL-like language (Stream Analytics), 235
SQL Server, 133
SQL Server AlwaysOn
    configured in a three-node cluster, 334
evaluating hybrid and Azure-hosted architectures for deployment, 306–316
design considerations, managing Azure resources, 310–312
scenarios dictating a hybrid scenario, 312–315
System Center components supported in Azure, 306–309
System Center deployment, 309–310
use by Azure Site Recovery, 337
system-level availability, 206

Table storage, 130
adding a new row to a table, 131
application logging stored in, 322
tasks, 194
task virtual machines (TVMs), 194
TCP
ports used by Hybrid Connections, 170
TDE (Transparent Data Encryption), SQL Server, 97
Team Foundation Server, 280
temporary drives, 26
tokens, 176
Tomcat on Linux, setting up Azure Websites, 264
Traffic Manager
automatic failover for critical applications, 206
using, 53–54
using to globally scale web applications, 253
websites deployed to multiple regions, exposing via single URL, 295
Transact-SQL (T-SQL), 296
transient errors, 207
Transparent Data Encryption (TDE), SQL Server, 97
trust, 65
Trust Center, 96
trusted subsystems, 68
trust-worthy computing, 6
TumblingWindow function, 235
Twitter
enabling SSO access to, using Azure AD Access Panel, 115
support by Azure ACS, 90
2-factor authentication (2FA)

2-factor authentication (2FA), 111

U

Unified Installer, 309
unplanned maintenance, 9
Update Domains, 207, 327
Upgrade Domains, 291
User administrator role (Azure AD), 100
User Matching Rules (AAD Sync), 82
User role (Azure AD), 100
users
  external, inviting to access Azure resources, 112
  self-service group management, 114
  self-service password resets, 113

V

VIP (virtual IP) addresses, 16
  using to deploy application updates, 328
Virtual Machine Manager, 306
  monitoring capabilities, 317
  volume and performance concerns, 311
virtual machines (VMs), 264
  assignment to Cloud Service, 16
Availability Set and autoscaling, 214
Availability Sets, 334
Azure monitoring of, 322
Azure Premium storage and large VM instances, 192
  comparison with App Service Web Apps and Cloud Services, 266
creating and selecting sizes, 24–30
  drives, images, and VHD files, 30
  pricing tiers and machine series, 26–29
  using data disks, 29–30
D-series and DS-series for HPC environment, 193
endpoints, 18–19
hosting websites, 263
identifying appropriate type and size for storage solutions, 137–139
images, managing, 31–32
Network Security Groups (NSGs) on, 101
  on Azure, multi-instance and single-instance, 8
  options for domain-joining Azure VMs and Cloud Services, 172–173
  scaling applications on, 40–43
    autoscale, 41–43
    load balancing, 40
  states, managing, 33–36
    Custom Script Extension and DSC, 34
    for larger deployments, 35
    VM extensions, 33
tailored for high-performance computing workloads, 191
Windows PowerShell commands for, 357–359
Virtual Network, 45, 101
  virtual networks (VNets)
    access control for VMs, 101
    access control lists (ACLs) and network security groups (NSGs), 18–22
      network ACLs, 20
      NSGs, 20
      VM endpoints, 18
    creating a cloud-only network, 13–18
      IP addresses, 16
      name resolution and DNS servers, 17
      using Azure Management Portal, 13–16
VIRTUAL_NETWORK tag, 21
virtual private networks (VPNs), 45
  authenticating through VPN and certificates, 310
  designing hybrid solutions with Virtual Network and
  ExpressRoute, 45–47
    ExpressRoute, 48
    Point-to-Site VPN, 46
    Site-to-Site VPN, 47
  identifying constraints for connectivity with, 172
  Multi-Site VPN, 50
  other hybrid solutions, 51
  vNet-to-vNet VPN, 49
Web Apps, 171
Visual Studio
  ADAL and, sample scenario, 71–73
  Application Insights with, 324
  creating web deployment packages, 271–273
  creating websites with, 254–256
directing web applications in Azure, 258
VM Agent, 33
VM extensions, 33
vNet-to-vNet VPN, 49
VPNs. See virtual private networks

W
web APIs, designing applications that use, 158–168
  implementing custom Web API, 159–160
  scaling using Azure App Service Web Apps, 161–163
  securing a web API, 165–167
  WebJobs, 163–165
web applications
  creating data-centric web applications, 230–233
    Azure CDN, 232
    Azure DocumentDB, 231
    Azure Search, 231
  map of Azure services and application features, 231
  sample architecture, 232
  designing Web Apps, 251–304
    deploying websites, 268–286
    designing for scalability and performance, 252–268
    designing websites for business continuity, 286–299
Web Apps. See App Service Web Apps
  Web Deploy tool, 272, 278–279
  WebJobs, 163–165
  Web Role, 215
  Web Roles, 263
websites. See also Azure Websites; App Service Web Apps
  creating using Microsoft Visual Studio, 254–256
    debugging, 257–259
  globally scaling in App Service Web Apps, 252–253
Windows Communication Foundation (WCF) services, 51
  Cloud Services providing access to, 263
  exposing securely using Service Bus Relay, 169
Windows Identity Foundation (WIF), 166
Windows Notification Services (WNS), 237
Windows Phone, 141
  adding Mobile Services Offline Sync capacity to apps, 146
  implementing Azure Mobile Services, 143
Windows PowerShell, 135, 354, 365
  Azure AD Module for, 79
  creating script specific to Azure, 354–363
    basics of Windows PowerShell, 356
  getting started with PowerShell, 354
  Windows PowerShell workflows, 360–361
  working with Storage, 359–360
  working with VMs, 357–359
Desired State Configuration, 34
  in Azure Automation, 369
PowerShell Deployment Toolkit, 310
  using for automation, 368
  using scripts to add Cloud Services to a domain, 172
Windows Server
  Configuration Manager services for, 314
Windows Server 2012
  Routing and Remote Access Service (RRAS), 47
Windows Server 2012 R2 VM, creating, 25–30
Windows Server Active Directory
  connecting Azure AD to, 71
Windows Server Failover Cluster (WSFC), 334
Windows Server Update Services (WSUS)
  use case for, 325
Windows Store Apps
  push notification for, in Mobile Services, 153
Windows systems, 141
  adding Mobile Services Offline Sync capacity to apps, 146
  Azure VMs on, 265
  data encryption features, 97
  implementing Azure Mobile Services, 143
Windows Update, 326
WinJS library, 144
Worker Role, 215
Worker Roles, 263
Workflows, 35
  work items, 194
  workload partitioning, 212
Workplace Join, 116
WSFC. See Windows Server Failover Cluster
ws-Federation

ws-Federation, 70
Azure ACS and, 90
WSUS (Windows Server Update Services)
use case for, 325

X

Xamarin, 143
xplat-cli. See Cross-Platform Command-Line Interface

Y

Yahoo!, support by Azure ACS, 90

Z

Zone-Redundant Storage (ZRS), 103
About the authors

HAISHI BAI, senior technical evangelist at Microsoft, focuses on the Microsoft Azure compute platform, including IaaS, PaaS, networking, and scalable computing services.

STEVE MAIER is an expert in Windows Store apps, mobile apps, and the cloud. He has been running a mobile phone community group for the past five years. He holds multiple certifications including Microsoft Specialist Architecting Microsoft Azure Solutions. You can reach Steve on his blog at http://42base13.net or on Twitter (@stevemaier3).

DAN STOLTS is a technology expert who is a master of systems management and security. You can reach him on his primary blog at http://itproguru.com or Twitter (@ITProGuru). He is proficient in many datacenter technologies (Windows Server, System Center, Virtualization, Cloud, and so on) and holds many certifications including MCT, MCITP, MCSE, and TS. Dan is currently specializing in system management, virtualization, and cloud technologies. He is and has been a very active member of the user group community. Dan is an enthusiastic advocate of technology and is passionate about helping others. To see more, go to http://itproguru.com/about.