

# Microsoft Azure Networking

The Definitive Guide



Avinash Valiramani

**FREE SAMPLE CHAPTER** |



# Microsoft Azure Networking: The Definitive Guide

Avinash Valiramani

# Microsoft Azure Networking: The Definitive Guide

Published with the authorization of Microsoft Corporation by:  
Pearson Education, Inc.

Copyright © 2023 by Pearson Education Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit [www.pearson.com/permissions](http://www.pearson.com/permissions).

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-13-756989-2

ISBN-10: 0-13-756989-0

Library of Congress Control Number: 2022938822

ScoutAutomatedPrintCode

## TRADEMARKS

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

## WARNING AND DISCLAIMER

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

## SPECIAL SALES

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com). For questions about sales outside the U.S., please contact [intlcs@pearson.com](mailto:intlcs@pearson.com).

## CREDITS

EDITOR-IN-CHIEF

Brett Bartow

EXECUTIVE EDITOR

Loretta Yates

SPONSORING EDITOR

Charvi Arora

DEVELOPMENT EDITOR

Kate Shoup

MANAGING EDITOR

Sandra Schroeder

SENIOR PROJECT EDITOR

Tracey Croom

COPY EDITOR

Sarah Kearns

INDEXER

Timothy Wright

PROOFREADER

Donna E. Mulder

TECHNICAL EDITOR

Thomas Palathra

EDITORIAL ASSISTANT

Cindy Teeters

COVER DESIGNER

Twist Creative, Seattle

COMPOSITOR

codeMantra

GRAPHICS

codeMantra

# Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content accurately reflects the histories and experiences of the learners we serve.
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview).

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

- Please contact us with concerns about any potential bias at <https://www.pearson.com/report-bias.html>.



# Contents at a Glance

	<i>About the author</i>	<i>xiii</i>
	<i>Acknowledgments</i>	<i>xv</i>
	<i>Introduction to Azure networking services</i>	<i>xvii</i>
<b>Chapter 1</b>	<b>Azure virtual networks</b>	<b>1</b>
<b>Chapter 2</b>	<b>Azure Application Gateway</b>	<b>15</b>
<b>Chapter 3</b>	<b>Azure VPN gateway</b>	<b>35</b>
<b>Chapter 4</b>	<b>Azure Load Balancer</b>	<b>55</b>
<b>Chapter 5</b>	<b>Azure Firewall</b>	<b>77</b>
<b>Chapter 6</b>	<b>Azure DNS</b>	<b>103</b>
<b>Chapter 7</b>	<b>Azure Traffic Manager</b>	<b>123</b>
<b>Chapter 8</b>	<b>Azure Front Door</b>	<b>145</b>
<b>Chapter 9</b>	<b>Azure Bastion</b>	<b>167</b>
<b>Chapter 10</b>	<b>Azure Private Link</b>	<b>183</b>
	<i>Index</i>	<i>201</i>



# Contents

	<i>About the author</i>	<i>xiii</i>
	<i>Acknowledgments</i>	<i>xv</i>
	<i>Introduction to Azure networking services</i>	<i>xvii</i>
<b>Chapter 1</b>	<b>Azure virtual networks</b>	<b>1</b>
	Overview .....	1
	Azure virtual networks features .....	1
	Design concepts and deployment considerations .....	2
	Address space	2
	Subnets	2
	vNET peering	2
	Routing	3
	Network security groups (NSGs)	6
	Availability zones support	6
	vNET network address translation (NAT)	7
	Integrations for enhanced security	7
	Service tags	7
	Disaster recovery planning	8
	vNET walkthrough	8
	Best practices .....	12
<b>Chapter 2</b>	<b>Azure Application Gateway</b>	<b>15</b>
	Overview .....	15
	Azure Application Gateway features .....	16
	Design concepts and deployment considerations .....	18
	Front-end IP addresses	19
	Back-end pools	20
	Listeners	20
	Request routing rules	21
	HTTP settings	22



	Health probes	23
	Sizing and scaling	23
	TLS policy	23
	Application gateway walkthrough	24
	Best practices .....	33
<b>Chapter 3</b>	<b>Azure VPN gateway</b>	<b>35</b>
	Overview .....	35
	Azure VPN gateway features .....	35
	Design concepts and deployment considerations .....	36
	VPN types	37
	Gateway SKUs	37
	Connection types	37
	Gateway subnet	38
	Border Gateway Protocol (BGP)	38
	Local network gateways	38
	VPN gateway redundancy	39
	Deployment models	39
	Best practices .....	54
<b>Chapter 4</b>	<b>Azure Load Balancer</b>	<b>55</b>
	Overview .....	55
	Azure Load Balancer features	56
	Design concepts and deployment considerations .....	56
	Front-end IP address	57
	Back-end pool	57
	Health probes	57
	Load-balancing rules	58
	Load-balancing algorithms	60
	Availability zones	61
	Azure Load Balancer walkthrough .....	62
	Best practices .....	74

<b>Chapter 5</b>	<b>Azure Firewall</b>	<b>77</b>
	Overview .....	77
	Azure Firewall features .....	78
	Design concepts and deployment considerations .....	79
	Support for availability zones	79
	Inbound DNAT	79
	Outbound SNAT	79
	Traffic filtering	80
	Groupings	81
	Forced tunneling	83
	Threat intelligence	83
	Azure Firewall Manager	83
	Classic rules versus firewall policies	84
	Firewall rule processing	84
	DNS proxy	86
	Active FTP support	86
	Azure Monitor logging	86
	Azure Firewall deployment walkthrough	87
	Best practices .....	100
<b>Chapter 6</b>	<b>Azure DNS</b>	<b>103</b>
	Overview .....	103
	Azure DNS features	103
	Azure DNS limitations	104
	Design and configuration considerations .....	105
	Types of DNS zones	105
	Linking with Azure virtual networks	109
	Auto registration	111
	Alias record sets	113
	Reverse DNS lookup	116
	Zone delegation	119
	Best practices .....	119

<b>Chapter 7</b>	<b>Azure Traffic Manager</b>	<b>123</b>
	Overview .....	123
	Traffic Manager features.....	123
	Design concepts and deployment considerations .....	124
	Traffic Manager endpoints	124
	Nested Traffic Manager profiles	126
	Traffic routing methods	127
	Endpoint monitoring	133
	Traffic Manager walkthrough	134
	Best practices .....	141
<b>Chapter 8</b>	<b>Azure Front Door</b>	<b>145</b>
	Overview .....	145
	Key features.....	145
	Design concepts and deployment considerations .....	146
	Back ends	147
	Back-end pools	147
	Health probes	148
	Load balancing	148
	Traffic routing	148
	URL rewrite	149
	URL redirect	149
	Wildcard domains	151
	Rules Engine	151
	Caching	151
	Network and security.....	153
	Azure DDoS Protection Basic	153
	Protection against unwanted protocols	153
	Handling large volumes of traffic	153
	WAF security features	154
	Front Door service walkthrough	154
	Best practices .....	165

<b>Chapter 9</b>	<b>Azure Bastion</b>	<b>167</b>
	Overview .....	167
	Azure Bastion features	167
	Azure Bastion limitations	168
	Design concepts and deployment considerations .....	169
	Architecture	169
	SKUs	171
	High-availability hosts	171
	Virtual network peering	171
	Disaster recovery	171
	Service requirements	172
	Best practices .....	179
<b>Chapter 10</b>	<b>Azure Private Link</b>	<b>183</b>
	Overview .....	183
	Azure Private Link features	183
	Design concepts and deployment considerations .....	184
	Private endpoints	184
	Azure Private Link service	192
	Best practices .....	200
	<i>Index</i>	<i>201</i>



# About the Author

---

**Avinash Valiramani** is an IT Infrastructure and Cloud Architect with more than 15 years of expertise in areas of Microsoft Technologies such as Microsoft Azure, Microsoft 365, Office365, Windows Server, Microsoft Exchange, SCCM, Intune, Hyper-V, and others. He is a certified Architect on Azure and Microsoft365 and primarily helps enterprises globally in their Cloud Roadmap Architecture and Onboarding/Migration Strategies & Implementation. Avinash is publishing four books on Microsoft Azure Best Practices series including this current one, collating real-world experiences to deliver a comprehensive and concise experience for new and budding technologists. Avinash also holds certifications in Barracuda, AWS, Citrix, VMware, and many other IT/Security industry certifications to complement his Microsoft expertise. He has authored a course of Azure Virtual Desktop for O'Reilly Media and is planning many others in the coming months. You can follow Avinash on Twitter at @avaliramani.



# Acknowledgments

---

I would like to thank Loretta Yates for trusting me with this huge responsibility. These books would not have been possible without your confidence in me and I will be forever grateful for that. I would like to thank Charvi Arora and the entire Microsoft Press/Pearson team for their constant support and guidance on this project. I would especially like to thank Kate Shoup for editing and reviewing this book and for all her guidance and attention to detail throughout these series of books. Kate, it has been a wonderful experience writing these four books with you and I could not have asked for a better collaborator. Thanks to Thomas Palathra for his thoughtful technical edits, Sarah Kearns for the amazing copy editing and Tracey Croom for adding the final touches to bring this to fruition. This book is the fruit of all our labor, and I am extremely happy we worked together on it.

I would also like to thank my family with gratitude, especially my brother Junaid and uncle Chandru on this effort. Your assistance in helping me organize my life and ensuring I could stay on track while wearing multiple hats was invaluable. This has been the biggest reason I managed to get this mammoth series of books out and it would not have been possible without all your support during this process. I would like to thank my mom for all her strength and belief throughout the years even when things were not going well and for believing in me throughout. Love you all.





# Introduction to Azure networking services

---

Welcome to *Microsoft Azure Networking: The Definitive Guide*. This book was developed to convey in-depth information about various Azure services that provide networking capabilities, as well as best practices based on real-life experiences using the product in different environments. The book is largely based on the versions of Azure networking services available during 2021 and early 2022, and takes into account the development work done on these services over the years. At that time, there were a few features and functionalities under preview. Because these features could change before becoming available to the general public, the most notable ones will be covered in subsequent iterations of this book, as they become available globally.

## Who is this book for?

---

*Microsoft Azure Networking: The Definitive Guide* is for anyone interested in Azure infrastructure solutions—not just IT and cloud administrators, network professionals, security professionals, developers, and engineers, but the entire spectrum of Azure users. Whether you have basic experience using Azure or other on-premises or cloud virtualization technologies or you are an expert, you can still derive value from this book. It provides introductory, intermediate, and advanced coverage of each networking service.

The book especially targets those who work in medium to large enterprise organizations and have at least one year of experience in designing, administering, deploying, managing, monitoring, and migrating network infrastructure to services such as Azure virtual networks, Azure Firewall, Azure Web Application Firewall, and others that comprise the Azure network stack.

## How is this book organized?

---

This book is organized into ten chapters:

- Chapter 1: Azure virtual networks
- Chapter 2: Azure Application Gateway

- Chapter 3: Azure VPN gateway
- Chapter 4: Azure Load Balancer
- Chapter 5: Azure Firewall
- Chapter 6: Azure DNS
- Chapter 7: Azure Traffic Manager
- Chapter 8: Azure Front Door
- Chapter 9: Azure Bastion
- Chapter 10: Azure Private Link

Each chapter focuses on a specific Azure networking service, covering the inner workings of each one in depth, walking you through how to build and test the service, and offering real-world best practices to help you maximize your Azure investment.

The approach adopted for this book is a unique mix of didactic, narrative, and experiential instruction:

- Didactic instruction covers the core introductions to the services.
- Narrative instruction leverages what you already understand to help you bridge that knowledge with new concepts introduced in the book.
- Experiential instruction takes into account real-world experiences and challenges facing small and large environments, as well as what factors to consider when designing and implementing workloads. Guided step-by-step walkthroughs show you how to configure each Azure networking service and its related features and options to gain all the benefits each service has to offer.

## System requirements

---

This book is designed to be tested using an Azure subscription. Microsoft offers a 30-day, \$200 USD trial subscription that you can use to test most services covered in this book. However, some services, such as dedicated hosts, cannot be used with a trial subscription. Testing and validating these services requires a paid subscription.

The following list details the minimum system requirements needed to use the content provided on the book's companion website:

- Windows 10/11 with the latest updates from Microsoft Update Service
- Azure PowerShell (<https://docs.microsoft.com/en-us/powershell/azure/install-az-ps>)

- Azure CLI (<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>)
- Display monitor capable of 1024 x 768 resolution
- Microsoft mouse or compatible pointing device

## About the companion content

---

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

*[MicrosoftPressStore.com/AzureNetworkingTDG/downloads](https://MicrosoftPressStore.com/AzureNetworkingTDG/downloads)*

or

*<https://github.com/avinashvaliramani/AzureNetworkingTDG>*

## 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:

*[MicrosoftPressStore.com/AzureNetworkingTDG/errata](https://MicrosoftPressStore.com/AzureNetworkingTDG/errata)*

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit *[MicrosoftPressStore.com/Support](https://MicrosoftPressStore.com/Support)*.

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>*.

## Stay in touch

---

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

## Overview

---

Over the years, Microsoft has introduced various services related to the Azure networking stack alongside the Azure compute services designed to leverage them. Microsoft has enhanced these services on a regular basis, making them more robust and resilient as well as easier to deploy and manage. The first of these services was Azure virtual machines (VMs). After that came additional platform as a service (PaaS) solutions like Azure App Service, Azure Container Service, Azure Functions, and Azure Virtual Desktop.

Following is a brief timeline of the announcement of each of these services in public preview:

- **Azure Traffic Manager** Nov 2013
- **Azure VPN Gateways** Dec 2014
- **Azure Load Balancer** Sept 2015
- **Azure Firewall** Nov 2015
- **Azure Application Gateway** Sept 2016
- **Azure DNS** Sept 2016
- **Azure Front Door** April 2019
- **Azure Bastion** Nov 2019
- **Azure Private Link** Feb 2020

Over the years, each service has added new capabilities to Azure's networking stack. These have provided customers with various networking-service options for use based on their application and security requirements.

Each service helps address different requirements in an organization's application design and architecture as well as overall security requirement. Each chapter of this book covers a single service, enabling you to dive into each one to better understand how it works and includes the associated best practices.

Each chapter initially focuses on factors to consider when selecting a particular networking service. Thereafter, it conveys in-depth concepts related to each service and the components that make up that service. This enables you to better understand how each service works. Once you have gained this understanding, you will focus on deployment considerations and strategies, with step-by-step walkthroughs of deployment methods, followed by best practices.

## Cloud service categories

---

As in other books in this series, let us start by first presenting the various cloud-service categories. Currently, cloud services are broken down into four main categories: infrastructure as a service (IaaS), platform as a service (PaaS), function as a service (FaaS), and software as a service (SaaS). SaaS is not relevant to the content covered in this book series, so the following explanations relate to the first three categories:

- **Infrastructure as a service (IaaS)** Using VMs with storage and networking is generally referred to as IaaS. This is a traditional approach to using cloud services in line with on-premises workloads. Most on-premises environments use virtualization technologies such as Hyper-V to virtualize Windows and Linux workloads. Migrating to IaaS from such an environment is a much easier first step than migrating to PaaS or FaaS. Over time, as an organization's understanding of various other types of cloud services grows, it can migrate to PaaS or FaaS.
- **Platform as a service (PaaS)** One of the biggest benefits of using a cloud service is the capability to offload the management of back-end infrastructure to the service provider. This model is called platform as a service (PaaS). Examples of back-end infrastructure include the various layers of an application, such as the compute layer, storage layer, networking layer, security layer, and monitoring layer. Organizations can use PaaS to free up their IT staff to focus on higher-level tasks and core organizational needs instead of on routine infrastructure monitoring, upgrade, and maintenance activities. Azure App Service and Azure Container Service are examples of Azure PaaS offerings.
- **Function as a service (FaaS)** These offerings go one step beyond PaaS to enable organizations to focus only on their application code, leaving the entire back-end infrastructure deployment and management to the cloud service provider. This enables developers to deploy their code without worrying about back-end infrastructure deployment, scaling, and management. It also enables the use of microservices architectures for applications. An example of an Azure FaaS offering is Azure Functions.

In the Azure networking stack, the services largely fall under the PaaS category. For example:

- Azure Firewall is a PaaS service that allows you to deploy a native firewall in Azure to protect both IaaS and PaaS workloads.

- Azure Bastion is a PaaS service that gives you the ability to securely access IaaS VM workloads in Azure using a browser without exposing them directly to the internet.

Each of these cloud service categories has various features and limitations. Limitations might relate to the application, technological know-how, and costs for redevelopment, among others. As a result, most organizations use some combination of various types of cloud services to maximize their cloud investments.

Each service provides a different level of control and ease of management. For example:

- IaaS provides maximum control and flexibility in migration and use.
- FaaS provides maximum automation for workload deployment, management, and use.
- PaaS provides a mix of both at varying levels, depending on the PaaS service used.

Each service also offers varying levels of scalability. For example:

- IaaS requires the use of additional services to achieve true scalability and load balancing—for example, using Azure Load Balancer, a PaaS service, to balance requests across multiple Azure IaaS VMs.
- PaaS and FaaS services are generally designed with built-in scalability and load-balancing features.

Cost-wise, each service provides varying levels of efficiency. For example:

- FaaS offerings charge for compute based only on the usage hours for compute services, making it extremely cost-effective.
- IaaS products charge for compute services regardless of usage once the compute service (for example, a VM) is online.
- PaaS offerings are a mixed bag depending on how the service is configured. Some PaaS products charge for the service regardless of usage, while others, if configured correctly, charge based on usage. For example, Azure Bastion has a fixed monthly cost for the service whereas Azure DNS is charged based on number of domains and number of queries per month.

## Service selection factors and strategies

---

There are certain factors to consider when selecting which Azure networking service would be ideal for a given environment based on the application architecture, connectivity requirements, application security requirements, application delivery requirements, and other business needs. Some of these key factors, and the Azure networking services that best addresses them, are as follows:

- **Deliver applications securely** The networking stack provides multiple services that you can leverage to securely deliver applications to your end users. These include Azure Front Door, Azure Traffic Manager, and Azure Load Balancer.
- **Protect application connectivity** You can protect connectivity to the applications using services such as Azure Firewall, Azure Private Link, Azure Web Application Firewall, and Azure Load Balancer. You can use these services individually or in combination to provide higher levels of protection.
- **Provide connectivity to Azure and on-premise resources** Services such as Azure virtual networks, Azure VPN Gateway, Azure vNET Peering, ExpressRoute, Azure Bastion, and Azure DNS provide you with different connectivity options to securely connect your Azure services to each other and to on-premise hosted services.

As you can see, there are multiple services for each factor. As you get more clarity on your requirements and a better understanding of each of these services, it will become clearer to you when each of these services should be used in your environment, as each one provides distinct capabilities.

## Selecting the right load-balancing service

---

Certain network load-balancing services provide functionality that is similar or overlapping in nature, such as Azure Front Door, Azure Traffic Manager, Azure Load Balancer, and Azure Web Application Gateway. Let us take a moment to narrow down which of these load-balancing services might be best suited for your application. Figure I-1 offers a good starting point for identifying which service might best serve your requirements. While the diagram shown in the figure is not exhaustive, it can help you narrow down which services to focus on before making your final decision.



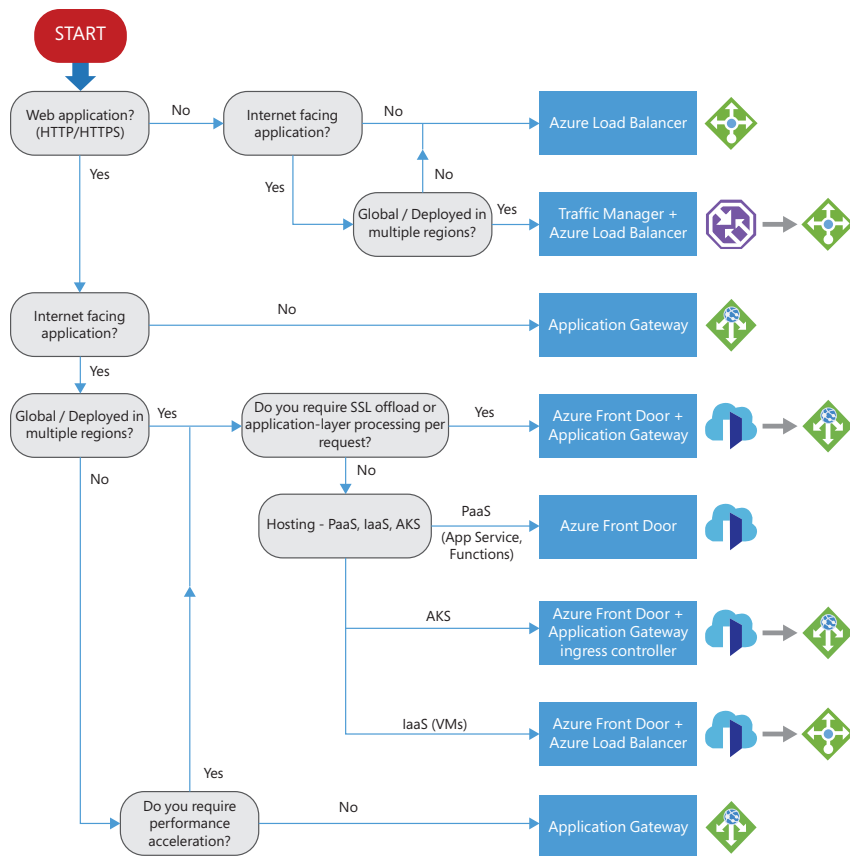


FIGURE I-1 Load-balancing service selection considerations.

Let us examine the flowchart shown in Figure I-1 in more detail.

- Non-web application workloads** The first factor you should consider is whether your workload is web application—published using HTTP/HTTPS. If not, then the Azure Load Balancer service might be best suited to handle your needs. If your application is hosted in multiple Azure regions with local redundancy in each region, a combination of Azure Traffic Manager and the Azure Load Balancer might be best suited to meet those needs.
- Web application workloads** When it comes to web application workloads, you need to consider a few factors before you can identify the appropriate service:

- **Will the web app be accessible only internally?** If it will only be accessible internally, then you can use the Azure Application Gateway service. If, however, the web app will be publicly accessible, then you'll need to consider the following factors as well.
- **Will the web application be hosted in multiple Azure regions?** If you plan to host the application in a single Azure region, then Azure Application Gateway might suffice (unless you need to accelerate application performance). If you will be hosting the web app in multiple Azure regions, then there are multiple Azure networking services that you can choose from, including Azure Front Door, Azure Application Gateway, Azure Load Balancer, or a combination of these. To better narrow down the appropriate option, let us continue further down the chain.
- **Will the multi-region public web application require SSL offloading or application request processing?** If this is a requirement, then the Azure Front Door service might be most appropriate for your needs. If, however, you do not require SSL offloading or application request processing, then depending on the application hosting model (such as Azure Kubernetes Service, Azure App Service, Azure Functions, or Azure VMs), you can work with a combination of the Azure Front Door, Azure Application Gateway, and Azure Load Balancer services.

As you can see, different factors can affect your decision-making process. Moreover, these may evolve over time, as your application requirements evolve. You might start with an application hosted in a single region and over time move it to multiple regions for global scalability.

As you read this book, you will better understand how you can leverage these networking services as needed over time to meet your ongoing needs and business demands.

## Conclusion

---

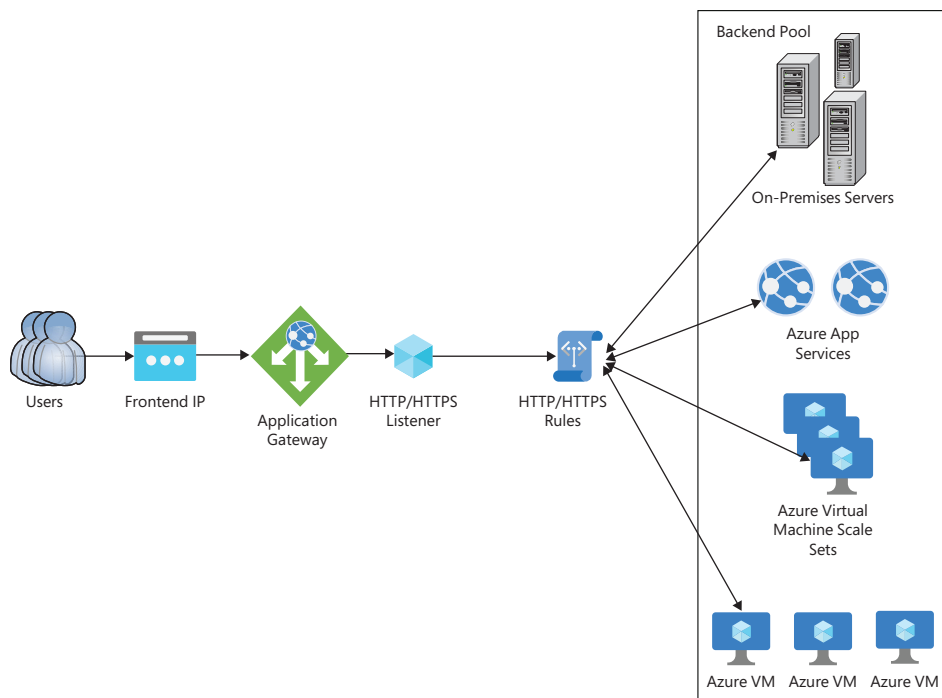
Now that you have an overview of the various Azure networking cloud offerings, let us dive deeper into each of the networking services. We'll start with the network service that forms the backbone of most Azure deployments: Azure virtual networks.



# Azure Application Gateway

## Overview

Azure Application Gateway is an ideal load balancer for web servers and applications that require HTTP/HTTPS traffic load-balancing and routing. It operates at the application layer (OSI layer 7 of the TCP and UDP stack) and can therefore analyze incoming traffic for custom routing based on URL paths and host headers in the incoming request. (See Figure 2-1.)



**FIGURE 2-1** The Azure Application Gateway handles incoming traffic using routing rules to back-end services.

Traditional load balancers operate at OSI layer 4, meaning they can only route traffic with limited parameters, such as the source IP or port and the destination IP and port. Due to these limitations, complex application traffic routing is difficult on traditional load balancers. In contrast, Azure Application Gateway can route traffic based on the URL in the header information

of the incoming traffic request. This makes it possible to route traffic for the same host header, directed to the same Azure Application Gateway IP, to a different server or server pool. For example, a request with the URL *www.contoso.com/videos* can be routed to one server or server pool hosting video-based content, a request for *www.contoso.com/images* can be routed to another server or server pool hosting image-based content, and so on. (See Figure 2-2.) This can help you design and optimize a web application server pool based on the content hosted by the pool.

App Gateway URL routing method

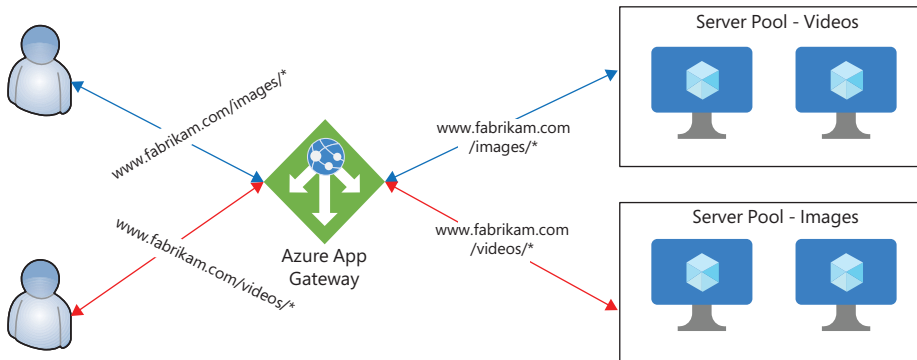


FIGURE 2-2 Azure Application Gateway URL-based routing.

You can deploy an Azure Application Gateway in one of two ways:

- **As an internal-only Azure Application Gateway** This is ideal for environments in which the application must be accessible only over internal virtual networks (vNETs). The gateway has an internal IP address, and the DNS records for it are set up in an internal or custom DNS server or service that is accessible only via vNETs for client connectivity.
- **As an internet-facing Azure Application Gateway** This has a public IP, which publicly exposes the gateway. So, back-end applications can be publicly accessed by clients by way of the internet. The DNS name for a public-facing load balancer is added to the public DNS by the Azure service.

## Azure Application Gateway features

The following list outlines the main features of the Azure Application Gateway. These features help explain how this service can be a key component of any web application design:

- **Web application firewall (WAF)** The Azure Application Gateway can act as a WAF, monitoring incoming traffic from clients and intercepting any malicious activities to provide centralized protection from well-known vulnerabilities for multiple web applications in your environment. You can host an application behind the WAF even if you have not yet been able to patch the vulnerabilities on the web servers themselves. You can centrally protect an application or even a large server farm that needs patching from a known vulnerability by patching the WAF. This serves as an interim solution until the application or server farm itself can be updated.

**NOTE** WAF rules are based on Open Web Application Security Project (OWASP) core rule sets 3.1 (WAF v2), 3.0 (WAF v1 and v2), and 2.2.9 (WAF v1 and v2).

- **Multiple-site hosting** Azure Application Gateway supports the use of multiple host, domain, or subdomain names on the same application gateway. So, you can use a single application gateway to handle web traffic for multiple web applications. This enables you to maximize your investment in the service by using each instance to its fullest capacity. You can map more than 100 web applications to a single application gateway. And, by using multi-site listeners, you can route traffic sent to a single public IP to different back-end server pools based on the URL request or host header.
- **Web-traffic redirection** By redirecting web traffic meant for one port to another, you can mask the application ports used internally on the web application servers, which can improve the security of your web applications. Traffic redirection also enables you to centrally route HTTP traffic to HTTPS, ensuring that no unencrypted communication occurs between your clients and web services. Web-traffic redirection supports the following scenarios:
  - **Global traffic redirection** This type of redirection enables you to redirect all traffic from HTTP to HTTPS on a site or to any other non-standard port required by the web application.
  - **Path-based redirection** With this type of redirection, you can redirect HTTP to HTTPS or any other non-standard port only on specific site areas, such as traffic to `/videos/*` or `/checkout/*`.
  - **Redirection to an external site** This type of redirection allows you to redirect traffic to an external site.
- **Secure Sockets Layer/Transport Layer Security (SSL/TLS) termination** You can use Azure Application Gateway to offload SSL/TLS processing for your web applications. This helps reduce the overhead of SSL/TLS encryption/decryption as well as server resource consumption. Traffic between the application gateway to the web application hosted internally behind it can be unencrypted. In some scenarios, however, you might need the back-end traffic to be encrypted for compliance or application-design reasons. Azure Application Gateway supports these types of end-to-end encryption scenarios, too.
- **Session affinity handling** You can set up Azure Application Gateway to support cookie-based session affinity, thereby ensuring that any sessions that are interrupted or dropped will reconnect to the same server as before. This can be a critical requirement for some applications where a user's session state is stored locally on the server.
- **Static virtual IP (VIP) assignment** The Standard\_v2 version of Azure Application Gateway supports the use of static VIP addresses, ensuring the VIP is maintained as-is for the lifetime of the application gateway.
- **Zone redundancy** You can set up Azure Application Gateway to span multiple availability zones, thereby improving the gateway's SLA and resiliency.

**NOTE** Currently, only the Standard\_v2 version of Azure Application Gateway supports zone redundancy.

- **Path-based routing** This enables you to analyze and route traffic based on the path indicated in the incoming web request. You can set up the application gateway to route traffic to different back-end servers or server pools based on the paths found in the request. So, content for different paths of a URL can be hosted on different servers or server pools, and the content itself can be optimized to deliver the best end-user experience possible.
- **Automatic scaling** You can set up Azure Application Gateway to automatically scale up or down based on traffic load at any given time. You need not select the “perfect” size for the application gateway when you provision it, because the gateway can scale as needed as traffic grows over time.
- **Support for WebSocket and HTTP/2 traffic** Azure Application Gateway natively supports WebSocket and HTTP/2 protocols. WebSocket is enabled by default and cannot be turned off. It allows full duplex communication between the web application server and client over long-running TCP connections, which can be optimized and used for multiple requests and responses. HTTP/2 protocols can be used only for client-to-application gateway communications. HTTP/2 is designed to function more efficiently than HTTP-based communications by eliminating the need for the constant polling required by HTTP to keep long-running sessions alive, which reduces the application gateway’s overhead with client communications. Both protocols are designed to work over ports 80 and 443, so you can easily incorporate them into an environment without making firewall changes.

## Design concepts and deployment considerations

---

Azure Application Gateway consists of a number of components that come together to filter traffic and provide secure routing services for your web applications. The main components of the Azure Application Gateway service are as follows (see Figure 2-3):

- Front-end IP addresses
- Back-end pools
- Listeners
- Request routing rules
- HTTP settings
- Health probes

It is important to have a solid understanding of each of these components to be able to design and deploy them appropriately based on your environment’s requirements. In addition,

you should have a strong grasp of sizing and scaling as well as TLS policy. This section covers all of these topics.

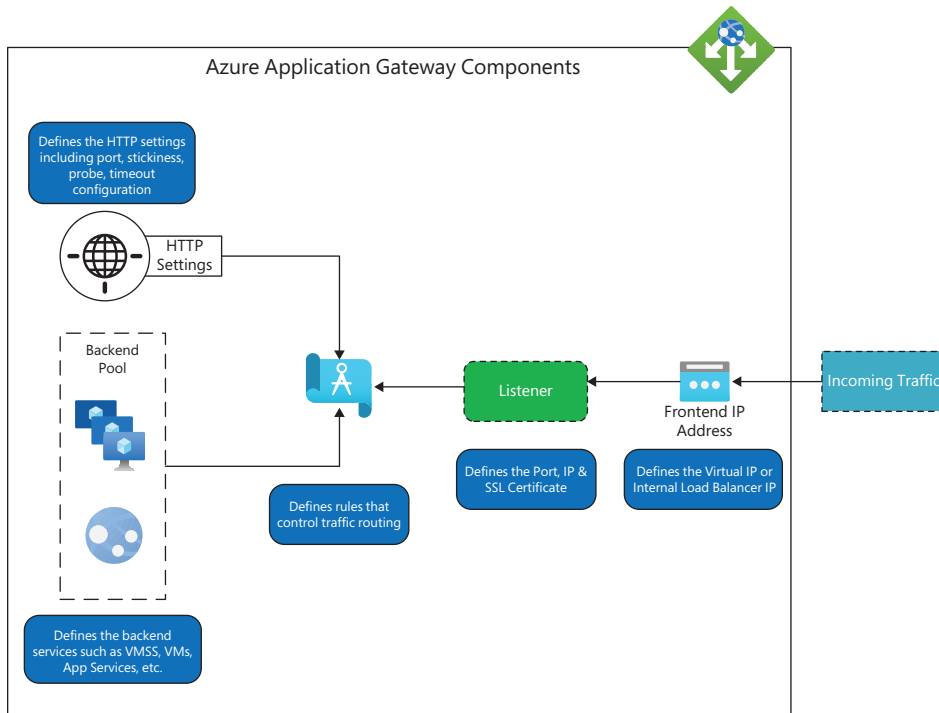


FIGURE 2-3 Azure Application Gateway components.

## Front-end IP addresses

A front-end IP address is the ingress point for the application gateway to receive web application traffic. It is referenced in the internal and/or public DNS and used to route application traffic. The front-end IP address can be an internal private IP address, a public IP address, or both.

If a private IP address is to be used, the gateway should be deployed in the same region as the vNET with which it is associated. Similarly, you should create any public IP that must be associated as the front-end IP address in the same region as the application gateway.

The application gateway supports both static and dynamic IP addresses for the private and public IP addresses. However, based on the SKU of the gateway, the support varies for each. SKU v1 supports dynamic public IP addresses and static or dynamic private IP addresses; there is no support for static public IP addresses. SKU v2 supports static and dynamic private and public IP addresses. You cannot set up a static private IP address along with a dynamic public IP address, but you can set up a static public IP address with a dynamic private IP address if required.



**NOTE** In general, if the gateway is stopped or started, the dynamic IP address changes. However, the dynamic IP address is retained if the gateway restarts due to failure or during upgrades or Azure host updates.

**PRO-TIP** Use the application gateway DNS name as a CNAME record in any public or internal DNS zones. This name does not change unless the application gateway is deleted.

## Back-end pools

Back-end pools are back-end servers or services to which web traffic requests from clients are routed. They serve the client request with the required application response via the application gateway.

Back-end pools support the following types of services:

- Virtual network interfaces
- Virtual machine scale sets (VMSS)
- Public IP addresses associated with Azure virtual machines (VMs) or services
- Internal IP addresses associated with Azure or on-premises servers or services
- Fully qualified domain names (FQDNs)
- Azure services, such as App Service

Back-end pools can be located across Azure regions, across different clusters, or outside Azure in a client's or third-party datacenter. As long as the application gateway can reach the back-end endpoint over TCP/IP, it can serve as a front-end to that web application.

For communications with on-premises servers, a VPN gateway or Azure ExpressRoute connection is necessary. For communication with Azure VMs and services over their private internal IPs, vNET peering or VPN gateway integration is needed.

**NOTE** A single application gateway can support different types of back-end pools to address different web application requirements and scenarios.

## Listeners

Every application gateway requires a minimum of one listener and can contain multiple listeners. A listener helps you define the protocol, port, hostname, and source IP address that is allowed to communicate with the back-end pool. Based on the configuration of the listeners defined on the application gateway, traffic is passed through or dropped. If the listener allows traffic through, the gateway evaluates it against routing rules configured to route traffic to the correct back-end pool.

There are two types of listeners:

- **Basic** A basic listener can listen for requests only for a single domain.
- **Multi-site** A multi-site listener can listen to requests for multiple host names or domain names. This type of listener can support more than 100 websites, which you can route to their own back-end pool.

Both types of listeners support the following ports:

- **SKU V1** Ports 1 – 65502
- **SKU V2** Ports 1 – 65199

In addition, they support the following protocols:

- HTTP
- HTTPS
- HTTP/2
- WebSocket

There are a few caveats to be aware of, as follows:

- WebSocket protocol support is enabled by default and cannot be turned off.
- HTTP/2 protocol support is disabled by default and must be turned on manually.
- HTTP/2 protocol support is limited to the client and application gateway communications. Any back-end communications occur over HTTP/1.1.

## Request routing rules

A request routing rule defines how traffic received by the application gateway should be routed. It binds the listener to a back-end server pool based on the HTTP settings defined to monitor in the request. One listener can be attached to only one rule.

Routing to the back-end pool depends on the rule configuration that defines which back-end pool binds to which URL or URL path. The routing rule also specifies whether any request must be rewritten before being routed to the back-end pool.

There are two types of request routing rules:

- **Basic** A basic request routing rule forwards all traffic to the associated back-end pool based on the HTTP settings associated with the rule.
- **Path-based** A path-based request routing rule analyzes the URL path in the request to identify the back-end pool to which to route the request. The rule contains different URL paths, set up to route to different back-end pools. If the incoming request does not match any of the rules, the traffic is routed to the default back-end pool based on associated HTTP settings.

A request routing rule is evaluated based on the priority assigned to that rule. By default, the priority is automatically assigned based on the order of rule creation (unless a specific priority is provided at the time of rule creation or set later on).

It is important to take priority into consideration in case there are rules that contain domains that overlap—for example, \*.fabrikam.com and blogs.fabrikam.com. In such cases, wildcard rules should be lower in priority to ensure the individual domain rules are evaluated before the wildcard rules. Otherwise, requests will be routed to the back-end pools associated with the wildcard domains only.

## Redirection support

In addition to routing traffic to back-end pools, request routing rules can be set up to redirect traffic to and from any port to a redirection target. The redirection could be to another listener or an external site. Redirection routing helps redirect HTTP traffic to HTTPS, or traffic from a standard web port (such as 80) to a non-standard port.

Azure Application Gateway supports different types of redirection:

- 310 Permanent Redirect
- 302 Found
- 303 See Other
- 307 Temporary Redirect

## Rewriting of HTTP headers and URLs

Azure Application Gateway allows HTTP request and response headers to be modified before the packet is sent to the back-end pool. So, the URL can be rewritten with custom security header fields, removing sensitive header information such as port information on X-Forwarded-For (XFF) headers. This feature can apply these changes only when certain conditions are met, so you can target rewrites to address any complex scenario as required.

## HTTP settings

HTTP settings define the back-end servers' port number, protocol, encryption settings, and other details. The application gateway uses these settings to route traffic to back-end servers when it receives a matching request.

HTTP settings are also used to define other settings, such as the following:

- **Cookie-Based Session Affinity** This setting instructs the gateway to use affinity to always route requests from the same client to the same hosts (assuming the host is online).
- **Connection Draining** This setting instructs the application gateway to gracefully drain connections on back-end servers, as they may be taken down for maintenance.
- **Custom Health Probe** This setting helps the gateway understand how it should validate the health of the back-end pool.

## Health probes

Monitoring the health of back-end pool instances is a critical function. It helps the service decide which back-end pool instances are healthy and usable for request routing and which ones need to be taken out of service to avoid application outages. Health probes provide the application gateway with the hostname, URL path, probe interval, and failed response limits to help it identify unhealthy back-end pool instances. The application gateway performs health monitoring by default. However, custom health probes help the gateway target the right parameters to evaluate instance health. Therefore, it is highly recommended to define custom health probes for each individual back-end pool.

## Sizing and scaling

The v1 SKU offers different gateway sizes:

- **Small** Appropriate for test and dev scenarios.
- **Medium** Appropriate for small environments with a few hundred users accessing a web application.
- **Large** Recommended for most enterprise or multi-site scenarios, to handle higher loads.

With the v2 SKU, autoscaling is available, eliminating the need for different gateway sizes. In this case, you use manual scaling or auto-scaling configurations to handle load instead of attempting to determine the right instance size at the outset.

## TLS policy

With Azure Application Gateway, you can offload SSL/TLS. This way, SSL connections from clients are terminated on the application gateway, and internal communications with the application back-end can be encrypted or unencrypted (thereby reducing overhead on instances of the application). If the back-end instances are set up to communicate with the application gateway in an unencrypted manner, application certificates can be deployed and managed only on the application gateway. This makes it easier to track, maintain, and update them.

A TLS policy defines the different TLS protocol versions and cipher suites to be used during a TLS handshake. The order of the ciphers specifies the order in which they are evaluated at the time of the handshake. There are two mechanisms to control this, as follows:

- **Using a predefined TLS policy** Every application gateway interface includes three predefined TLS policies—each one defined to support different TLS protocol versions and cipher suites. The names of the policies indicate the dates on which they were introduced; it's recommended that you use the newer ones.
- **Using a custom TLS policy** If a predefined policy does not meet your requirements, you can customize a policy to include the TLS protocols and cipher suites (and their priority) based on your needs.

With both predefined and custom TLS policies, SSL 2.0 and 3.0 are set to disabled, and you cannot override this. However, with a custom policy, you can set up any of the three TLS protocol versions (v1\_0, v1\_1, or v1\_2) as the minimum required version. You can also set up all three of these with no minimum requirement if needed.

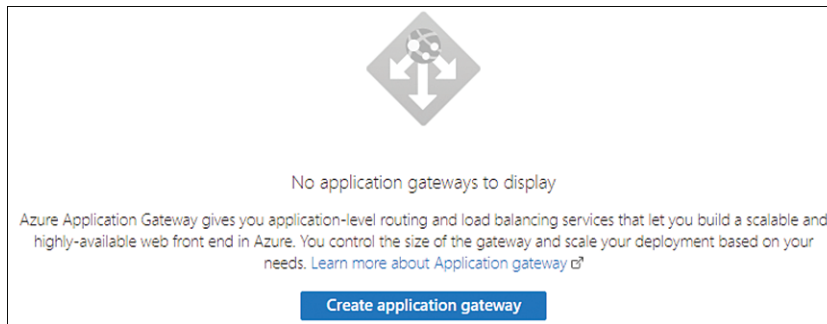
## Application gateway walkthrough

The following sections walk you through the process of creating an application gateway using the Azure Portal, Azure PowerShell, and the Azure CLI. If you are following along, be sure to select resources and resource names based on your environment, including a unique application gateway name for each of your deployments. Also be sure to delete any unwanted resources after you have completed testing to reduce charges levied by Microsoft for these resources. Finally, to complete this walkthrough, you need to have created at least two back-end VMs with IIS web services to integrate with the Application Gateway. Microsoft provides Windows Server templates in the Azure Marketplace that install and set up IIS with the default configuration; alternatively, you can create a custom configuration based on your requirements.

### USING THE AZURE PORTAL

To create an application gateway using the Azure Portal, follow these steps:

1. Log into the Azure Portal, type **application gateway** in the search box to locate the service, and select it from the list that appears.
2. Click **Create** or **Create Application Gateway** to start the Create Application Gateway wizard. (See Figure 2-4.)



**FIGURE 2-4** Click Create Application Gateway.

3. In the **Basics** tab of the Create Application Gateway wizard (see Figure 2-5), enter the following information and click **Next: Frontends**:
  - **Subscription** Select the subscription to host the application gateway.
  - **Resource Group** Select the resource group you want to host the application gateway. Alternatively, click the **Create New** link and follow the prompts to create a new resource group.

- **Name** Type a name for the application gateway. If the name you type is already in use, the wizard will prompt you to enter another name.
- **Region** Select the Azure region in which you want to create the application gateway.
- **Tier** Choose **Standard V2**.
- **Enable Autoscaling** Select the **Yes** option button.
- **Minimum Instance Count** Enter the minimum number of hosts to set up.
- **Maximum Instance Count** Enter the maximum number of hosts to which the gateway should scale.
- **Availability Zone** Leave this set to **None**.
- **HTTP2** Leave this set to **Disabled**.
- **Virtual Network** Select an existing vNET for which you have created back-end VMs.
- **Subnet** Select a subnet for the application gateway.

**Create application gateway** ...

1 Basics 2 Frontends 3 Backends 4 Configuration 5 Tags 6 Review + create

An application gateway is a web traffic load balancer that enables you to manage traffic to your web application. [Learn more about application gateway](#)

**Project details**  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

Application gateway name \*  ✓

Region \*

Tier

Enable autoscaling  Yes  No

Minimum instance count \*

Maximum instance count  ✓

Availability zone

HTTP2  Disabled  Enabled

**Configure virtual network**

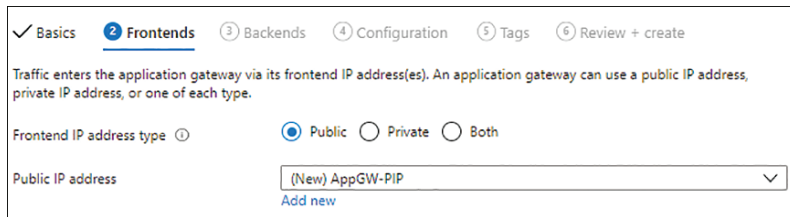
Virtual network \*  [Create new](#)

Subnet \*  [Manage subnet configuration](#)

Previous

**FIGURE 2-5** The Basics tab of the Create Application Gateway wizard.

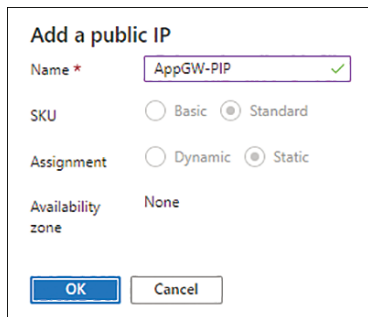
- In the **Frontends** tab of the Create Application Gateway wizard (see Figure 2-6), for **Frontend IP Address Type**, select the **Public** option button.



The screenshot shows the 'Frontends' tab of the 'Create Application Gateway' wizard. At the top, there are six steps: 1. Basics, 2. Frontends (selected), 3. Backends, 4. Configuration, 5. Tags, and 6. Review + create. Below the steps, a text box explains: 'Traffic enters the application gateway via its frontend IP address(es). An application gateway can use a public IP address, private IP address, or one of each type.' Underneath, there are three radio buttons for 'Frontend IP address type': 'Public' (selected), 'Private', and 'Both'. Below that is a dropdown menu for 'Public IP address' with the value '(New) AppGW-PIP' and an 'Add new' link.

**FIGURE 2-6** The Frontends tab of the Create Application Gateway wizard.

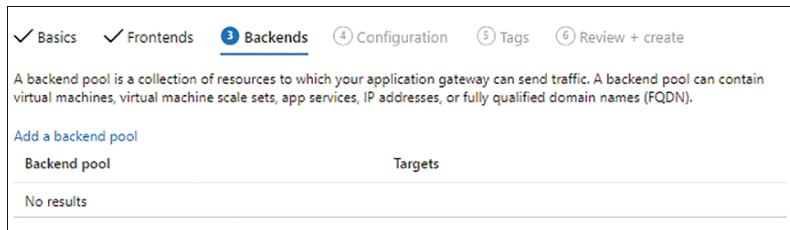
- Under the **Public IP Address** box, click the **Add New** link.
- On the **Add a Public IP** page, in the **Name** box, type a name for the public IP address. (See Figure 2-7.) Then click **OK**.



The screenshot shows the 'Add a public IP' dialog box. It has a title bar 'Add a public IP'. Below the title, there is a 'Name \*' field with the value 'AppGW-PIP' and a green checkmark. Below that are two radio buttons for 'SKU': 'Basic' and 'Standard' (selected). Below that are two radio buttons for 'Assignment': 'Dynamic' and 'Static' (selected). Below that is a text field for 'Availability zone' with the value 'None'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

**FIGURE 2-7** The Add a Public IP dialog box.

- In the **Frontends** tab of the Create Application Gateway wizard, click **Next: Backends**.
- In the **Backends** tab of the Create Application Gateway wizard (see Figure 2-8), click the **Add a Backend Pool** link.

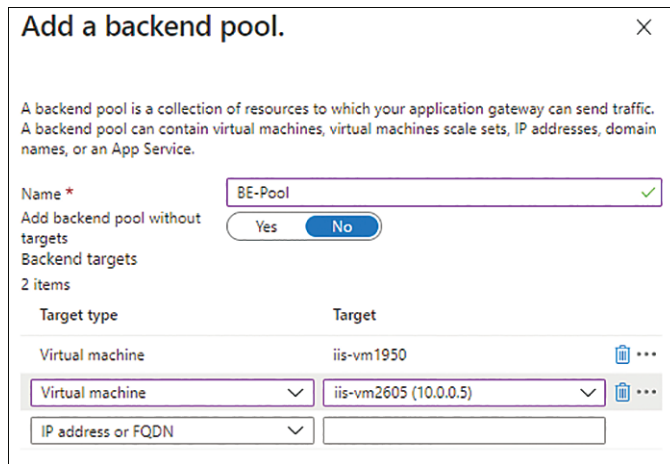


The screenshot shows the 'Backends' tab of the 'Create Application Gateway' wizard. At the top, there are six steps: 1. Basics, 2. Frontends, 3. Backends (selected), 4. Configuration, 5. Tags, and 6. Review + create. Below the steps, a text box explains: 'A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).' Below that is a link 'Add a backend pool'. Below the link is a table with two columns: 'Backend pool' and 'Targets'. The table is currently empty and shows 'No results'.

**FIGURE 2-8** The Backends tab of the Create Application Gateway wizard.

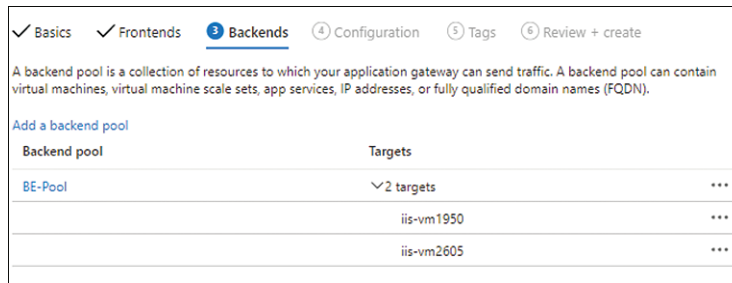
- In the **Add a Backend Pool** dialog box (see Figure 2-9), enter the following information and click **Add**:
  - Name** Type a name for the back-end pool.
  - Add Backend Pool Without Targets** Click **No**.

- **Target Type** Select **Virtual Machine**.
- **Target** Select one of the VMs you created for this walkthrough.



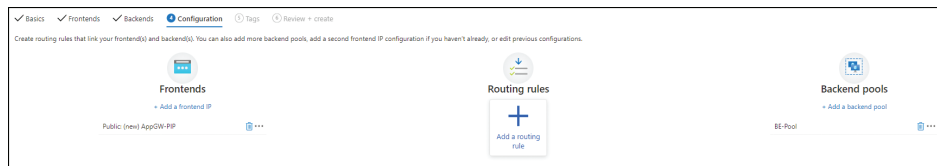
**FIGURE 2-9** The Add a Backend Pool dialog box.

10. In the **Backends** tab of the Create Application Gateway wizard (see Figure 2-10), verify that the back-end pool configuration is correct and click **Next: Configuration**.



**FIGURE 2-10** The updated Backends tab of the Create Application Gateway wizard.

11. In the **Configuration** tab of the Create Application Gateway wizard (see Figure 2-11), click **Add a Routing Rule** to create a routing rule for incoming traffic.



**FIGURE 2-11** The Configuration tab of the Create Application Gateway wizard.



12. In the **Listener** tab of the **Add a Routing Rule** settings (see Figure 2-12), enter the following information:
  - **Rule Name** Type a name for the rule.
  - **Listener Name** Type a name for the listener.
  - **Frontend IP** Select **Public**.
  - **Protocol** Select the protocol used by your application.
  - **Port** Type the port used by your application.
  - **Listener Type** Select the **Basic** option button.
  - **Error Page URL** Select the **No** option button.

**Add a routing rule** [X]

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name \*  ✓

\*Listener  Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener name \*  ✓

Frontend IP \*  ✓

Protocol  HTTP  HTTPS

Port \*  ✓

**Additional settings**

Listener type  Basic  Multi site

Error page url  Yes  No

**FIGURE 2-12** The Listener tab in the Add a Routing Rule settings.

13. Click the **Backend Targets** tab (look ahead to Figure 2-13).
14. For **Target Type**, select the **Backend Pool** option button.
15. Open the **Backend Target** drop-down list and select the back-end pool target.
16. Under **HTTP Settings**, click the **Add New link**.
17. In the **Add a HTTP Setting** settings (see Figure 2-14), enter the following information and click **Add**. (Leave the other settings as is.)
  - **HTTP Settings Name** Enter a name for the HTTP settings.
  - **Backend Protocol** Select the back-end protocol used by the application.
  - **Backend Port** Type the back-end port used by your application.
  - **Cookie-Based Affinity** Select the **Disable** option button.
  - **Connection Draining** Select the **Disable** option button.
  - **Request Time-Out (Seconds)** Enter a request time-out value (in seconds) that reflects the responsiveness of your application.

### Add a routing rule

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name \*

\* Listener \* **Backend targets**

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of HTTP settings that define the behavior of the routing rule.

Target type  Backend pool  Redirection

Backend target \*  [Add new](#)

HTTP settings \*  [Add new](#)

**Path-based routing**

You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of HTTP settings based on the URL path.

Path based rules

Path	Target name	HTTP setting name	Backend pool
No additional targets to display			

[Add multiple targets to create a path-based rule](#)

**FIGURE 2-13** The Backend Targets tab of the Add a Routing Rule settings.

### Add a HTTP setting

[← Discard changes and go back to routing rules](#)

HTTP settings name \*

Backend protocol  HTTP  HTTPS

Backend port \*

**Additional settings**

Cookie-based affinity  Enable  Disable

Connection draining  Enable  Disable

Request time-out (seconds) \*

Override backend path

**Host name**

By default, Application Gateway does not change the incoming HTTP host header from the client and sends the header unaltered to the backend. Multi-tenant services like App service or API management rely on a specific host header or SNI extension to resolve to the correct endpoint. Change these settings to overwrite the incoming HTTP host header.

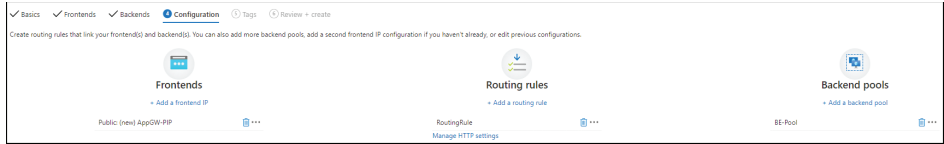
Override with new host name  Yes  No

Host name override  Pick host name from backend target  Override with specific domain name

Create custom probes  Yes  No

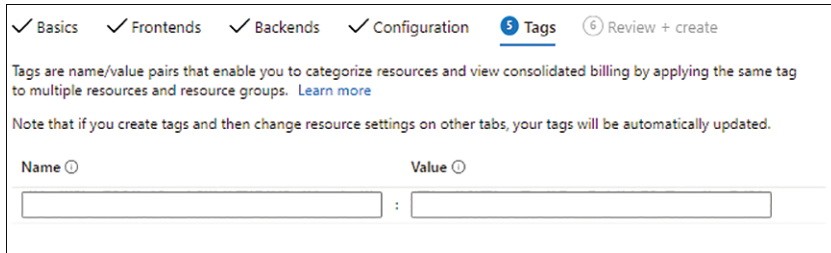
**FIGURE 2-14** The Add a HTTP Setting settings.

18. In the **Configurations** tab (see Figure 2-15), confirm that the configuration is correct and click **Next: Tags**.



**FIGURE 2-15** The updated Configuration tab of the Create Application Gateway wizard.

- In the **Tags** tab, enter any tags required for the application gateway or leave the fields blank (see Figure 2-16) and click **Next: Review + Create**.



**FIGURE 2-16** The Tags tab of the Create Application Gateway wizard.

- On the **Review + Create** tab (see Figure 2-17), review your settings, and click **Create**. You can now test whether your application is accessible by accessing the application gateway URL. (The URL will be visible in the application gateway's Overview tab.)



**FIGURE 2-17** The Review + Create tab of the Create Application Gateway wizard in the Azure Portal.

## USING AZURE POWERSHELL

You can create an application gateway using Azure PowerShell with the `New-AzVirtualNetworkGateway` and `New-AzVirtualNetworkGatewayConnection` commands and various switches to set the application gateway's parameters. The following code shows you how. Use this snippet to create the same application gateway as you did in the Azure Portal. (Replace all the variables and on-premises firewall configuration as per your environment.) When you do, be sure to either delete the previous application gateway or give this new application gateway a different name:

```
#Define variables
$agSubnetConfig = New-AzVirtualNetworkSubnetConfig `
    -Name myAGSubnet `
    -AddressPrefix 10.21.0.0/24
$backendSubnetConfig = New-AzVirtualNetworkSubnetConfig `
    -Name myBackendSubnet `
    -AddressPrefix 10.21.1.0/24
New-AzVirtualNetwork `
    -ResourceGroupName myResourceGroupAG `
    -Location eastus `
    -Name myVNet `
    -AddressPrefix 10.21.0.0/16 `
    -Subnet $agSubnetConfig, $backendSubnetConfig
New-AzPublicIpAddress `
    -ResourceGroupName myResourceGroupAG `
    -Location eastus `
    -Name myAGPublicIpAddress `
    -AllocationMethod Static `
    -Sku Standard
#Create IP Config and front-end port
$vnnet = Get-AzVirtualNetwork -ResourceGroupName myResourceGroupAG -Name myVNet
$subnet = Get-AzVirtualNetworkSubnetConfig -VirtualNetwork $vnnet -Name myAGSubnet
$pip = Get-AzPublicIpAddress -ResourceGroupName myResourceGroupAG -Name
myAGPublicIpAddress
$gipconfig = New-AzApplicationGatewayIPConfiguration `
    -Name myAGIPConfig `
    -Subnet $subnet
$fipconfig = New-AzApplicationGatewayFrontendIPConfig `
    -Name myAGFrontendIPConfig `
    -PublicIpAddress $pip
$frontendport = New-AzApplicationGatewayFrontendPort `
    -Name myFrontendPort `
    -Port 80
#Create backend pool
$backendPool = New-AzApplicationGatewayBackendAddressPool `
    -Name myAGBackendPool
$poolSettings = New-AzApplicationGatewayBackendHttpSetting `
```

```

-Name myPoolSettings `
-Port 80 `
-Protocol Http `
-CookieBasedAffinity Enabled `
-RequestTimeout 30
#Create listener and add rule
$defaultlistener = New-AzApplicationGatewayHttpListener `
-Name myAGListener `
-Protocol Http `
-FrontendIPConfiguration $fipconfig `
-FrontendPort $frontendport
$frontendrule = New-AzApplicationGatewayRequestRoutingRule `
-Name rule1 `
-RuleType Basic `
-HttpListener $defaultlistener `
-BackendAddressPool $backendPool `
-BackendHttpSettings $poolSettings
#Create application gateway
$sku = New-AzApplicationGatewaySku `
-Name Standard_v2 `
-Tier Standard_v2 `
-Capacity 2
New-AzApplicationGateway `
-Name myAppGateway `
-ResourceGroupName myResourceGroupAG `
-Location eastus `
-BackendAddressPools $backendPool `
-BackendHttpSettingsCollection $poolSettings `
-FrontendIpConfigurations $fipconfig `
-GatewayIpConfigurations $gipconfig `
-FrontendPorts $frontendport `
-HttpListeners $defaultlistener `
-RequestRoutingRules $frontendrule `
-Sku $sku

```

## USING THE AZURE CLI

You can create an application gateway using the Azure CLI with the `az network vnet-gateway create` command and various switches to set the application gateway's parameters. The following Bash script shows you how. Use this snippet to create the same application gateway as you did in the Azure Portal. (Replace all the variables and on-premises firewall configuration as per your environment.) When you do, be sure to either delete the previous application gateway or give this new application gateway a different name:

```

#Define variables
#Create network
az network vnet create \

```

```

--name myVNet \
--resource-group myResourceGroupAG \
--location eastus \
--address-prefix 10.21.0.0/16 \
--subnet-name myAGSubnet \
--subnet-prefix 10.21.0.0/24
az network vnet subnet create \
--name myBackendSubnet \
--resource-group myResourceGroupAG \
--vnet-name myVNet \
--address-prefix 10.21.1.0/24
az network public-ip create \
--resource-group myResourceGroupAG \
--name myAGPublicIPAddress \
--allocation-method Static \
--sku Standard
#Create application gateway
address1=$(az network nic show --name myNic1 --resource-group myResourceGroupAG | grep
"\privateIpAddress\:" | grep -oE '[^ ]+$' | tr -d '(',')')
address2=$(az network nic show --name myNic2 --resource-group myResourceGroupAG | grep
"\privateIpAddress\:" | grep -oE '[^ ]+$' | tr -d '(',')')
az network application-gateway create \
--name myAppGateway \
--location eastus \
--resource-group myResourceGroupAG \
--capacity 2 \
--sku Standard_v2 \
--public-ip-address myAGPublicIPAddress \
--vnet-name myVNet \
--subnet myAGSubnet \
--servers "$address1" "$address2"

```

## Best practices

---

Following are some recommended practices for deploying and managing your application gateway environment. These practices can help you make the most of your investment in this service:

- Monitor and plan instance count to avoid resource crunch** Azure Application Gateway v2 supports auto-scaling, but if you are using Azure Application Gateway v1, you will have to scale manually. The v1 SKU supports scaling up to 32 instances. To identify the right instance count, monitor CPU utilization for the application gateway for at least a month and identify the peak CPU usage. Then add a buffer of 15% to 20% to handle unexpected spikes and growth. Finally, select your instance count based on this sizing.

- **Upgrade to the v2 SKU as soon as possible** As mentioned, Azure Application gateway v1 supports manual scaling only—which is not the most efficient or cost-effective way to manage your gateway instances. In addition to auto-scaling, the v2 SKU offers other performance benefits, including SSL/TLS offloading, improved deployment performance, zone redundancy, and many others. You should upgrade to the v2 SKU as soon as you can so you can benefit from all these (and other) features.
- **Set the maximum instance count in the v2 SKU** Because the v2 SKU supports auto-scaling, and because charges are levied based on how many units are used, it is important to consider budgetary requirements when setting the maximum instance count. The v2 SKU supports a maximum of 125 instances—which means unplanned spikes can result in more instances being activated than are budgeted for.
- **Size the gateway subnet for future growth** Size the subnet in which you plan to host application gateway instances to take into account future scalability requirements. Changes to the subnet configuration are not currently supported and require a redeployment of the service, resulting in possible disruption.
- **Set the minimum instance count for v2 SKU** Bringing additional instances online to accept traffic when auto-scaling does take some time—usually between six and seven minutes. Any unexpected spikes during this period can result in traffic drops or higher response latency. You should monitor CPU usage for at least a month to identify the minimum instances required, and maintain a buffer of 15% to 20% to allow for unexpected spikes.
- **Monitoring and alerting** Set up alerts for different gateway metrics to monitor CPU usage, instance scaling, and network utilization so you can be notified of any anomalies that might cause potential outages. Examples of alerts could include average CPU usage spiking by 75% to 80% for a sustained period of time, too many failed requests, gateway not responding, logs containing numerous 4xx or 5xx errors (indicating response issues), and too many unhealthy back-end hosts.
- **Set up geo-filtering to block unwanted countries/regions** The v2 SKU supports geo-filtering, which enables you to allow or block traffic from specific countries or regions. It is a good practice to use this feature to prevent (or allow) traffic from certain locales to connect to your web applications to reduce your attack surface.
- **Set up bot protection to prevent attacks** The v2 SKU has a feature to prevent traffic from known bot networks. Enable this feature to intercept known malicious traffic before it reaches your web applications.
- **Set up diagnostics logging and long-term retention** Collect firewall, performance, and access logs for your application gateway instances and save them in Azure storage, Log Analytics, or an event stream. These logs can help you identify potential issues and take proactive action. Set up retention policies based on historical data storage comparison and compliance requirements for your organization.
- **Set up the latest TLS policy version for extra security** Use the latest TLS policy version (currently AppGwSslPolicy20170401S) to enforce TLS 1.2 and stronger ciphers.

# Index

## A

- access approvals, Private Link service, 195
- active FTP, Azure Firewall and, 86
- active-active mode, Azure VPN gateway, 41–42
- Activity logs, 87
- address space, 2, 12
- algorithms, load balancing, 60–61
- alias record sets, 113–116
- AnyCast networking, 103
- application FQDN traffic filtering, 81
- architecture, Azure Bastion, 169–170
- authentication
  - multi-factor, 180
  - point-to-site VPN gateway connections, 43–44
- auto registration, 111–112
  - using Azure CLI, 112
  - using Azure Portal, 112
  - using Azure PowerShell, 112
- autoscaling, 23. *See also* scaling
- availability zones, 6, 17, 36
  - Azure Firewall and, 79
  - Azure Load Balancer and, 61–62
- az network front-door create command, 163–165
- az network lb create command, 72–73
- az network private-dns link create command, 111
- az network private-endpoint create command, 191–192, 199
- az network traffic-manager profile create command, 140–141
- az network vnet command, 12
- Azure Activity Logs, 14
- Azure AD (Active Directory), 179
- Azure Application Gateway, 15–16
  - best practices, 33–34
  - components
    - back-end pools, 20
    - front-end IP addresses, 19–20
    - health probes, 23
    - HTTP settings, 22
    - listeners, 20–21
    - request routing rules, 21–22
  - creating
    - using Azure CLI, 32–33
    - using Azure Portal, 24–30
    - using Azure PowerShell, 31–32
  - deployment options, 16
  - features, 16–18
  - sizing and scaling, 23
  - TLS policy, 23–24
- Azure Bastion, 167
  - architecture, 169–170
  - best practices, 179–181
  - connecting to a VM using Azure Portal, 176–179
  - creating
    - using Azure CLI, 175–176
    - using Azure Portal, 172–175
    - using Azure PowerShell, 175–176
  - disaster recovery, 171
  - features, 167–168
  - limitations, 168–169
  - peering, 171
  - permissions, 172
  - redundancy, 171
  - service requirements, 172
  - SKUs, 171
- Azure CLI. *See also* commands
  - application gateway, creating, 32–33
  - Azure Bastion service, creating, 175–176
  - Azure Front Door service, setting up, 163–165
  - Azure Load Balancer, creating, 72–73
  - Azure Traffic Manager, setting up, 140–141
  - Azure VPN gateway, creating, 53
  - creating a vNET, 12
  - DNS zones, creating, 108
  - linking a private DNS zone to a vNET, 111
  - private endpoints, creating, 191–192



- Private Link service, creating, 199
- reverse DNS lookups, creating, 118
- setting up auto registration, 112
- Azure DNS, 103. *See also* DNS zones
  - best practices, 119–121
  - DNS zones, 105
    - private, 105
    - public, 105
  - features, 103–104
  - limitations, 104
  - queries, 119
  - RBAC and, 119–120
  - resource locks, 120–121
  - reverse DNS lookups, creating, 116
    - using Azure CLI, 118
    - using Azure Portal, 116–118
    - using Azure PowerShell, 118
  - zone delegation, 119
- Azure endpoints, 125. *See also* endpoints
- Azure Firewall, 13, 77
  - active FTP support, 86
  - best practices, 100–102
  - classic rules, 84
  - creating
    - using Azure Portal, 87–99
    - using Azure PowerShell, 99–100
  - DNAT (destination NAT), 79
  - DNS proxy, 86
  - features, 78
  - forced tunneling, 83
  - infrastructure FQDNs, 82
  - IP groups, 82
  - logging, 86
    - Activity logs, 87
    - diagnostic logs, 87
    - firewall metrics, 86
  - Manager, 83
  - policies, 84
  - rule processing
    - for incoming traffic, 84–85
    - for outbound traffic, 85
  - SNAT (source NAT), 79–80
  - tags, 81
    - FQDN, 81
    - service, 82
  - Threat Intelligence, 83
  - traffic filtering, 80
    - application FQDN, 81
    - network, 80
    - web, 80
  - web categories, 82
- Azure Front Door, 145
  - back ends, 147
  - back-end pools, 147
  - best practices, 165–166
  - caching, 151–153
  - dynamic content
    - compression, 152
  - features, 145–146
  - handling large volumes of traffic, 153
  - health probes, 148
  - integration with Azure DDoS Protection Basic, 153
  - logging, 165
  - performance counters, 165
  - protection against unwanted protocols, 153
  - query strings, 152
  - Rules Engine, 151
  - setting up
    - using Azure CLI, 163–165
    - using Azure Portal, 154–162
    - using Azure PowerShell, 162–163
  - traffic routing, 148–149
  - URL redirect, 149–150
  - URL rewrite, 149
  - WAF security features, 154
  - wildcard domains, 151
- Azure Load Balancer, 55
  - algorithms, 60–61
  - availability zones, 61–62
  - best practices, 74–75
  - components
    - back-end pool, 57
    - front-end IP addresses, 57
    - health probes, 57–58
  - creating
    - using Azure CLI, 72–73
    - using Azure Portal, 62–70
    - using Azure PowerShell, 71–72
  - deployment considerations, 56
  - features, 56
  - rules, 58
    - HA ports, 59
    - inbound NAT, 59
    - outbound SNAT, 60
- Azure Log Analytics, 13
- Azure Monitor, 14, 143–144
  - Azure Front Door and, 165
  - logging, 86
    - Activity logs, 87
    - diagnostic logs, 87
    - firewall metrics, 86
- Azure Policy, 14, 143, 166, 181
- Azure Portal
  - alias record sets, creating, 113–116
  - application gateway, creating, 24–30
  - Azure Bastion service, creating, 172–175
  - Azure Firewall, creating, 87–99
  - Azure Front Door service, setting up, 154–162
  - Azure Load Balancer, creating, 62–70
  - Azure Traffic Manager, setting up, 134–139
  - Azure VPN gateway, creating, 46–52
  - creating a vNET, 8–11
  - DNS zones, creating, 106–108

- linking a private DNS zone to a vNET, 110
- private endpoints, creating, 187–190
- Private Link service, creating, 195–198
- Real User Measurements, enabling, 141–143
- reverse DNS lookups, creating, 116–118
- setting up auto registration, 112
- using Azure Bastion to connect to a VM, 176–179
- Azure PowerShell
  - application gateway, creating, 31–32
  - Azure Bastion service, creating, 175–176
  - Azure Front Door service, setting up, 162–163
  - Azure Load Balancer, creating, 71–72
  - Azure Traffic Manager, setting up, 139–140
  - Azure VPN gateway, creating, 52–53
- commands
  - New-AzBastion, 175–176
  - New-AzDNSZone, 118
  - New-AzFirewall, 99–100
  - New-AzFrontDoor, 162–163
  - New-AzLoad Balancer, 71–72
  - New-AzPrivateDNSVirtualNetworkLink, 112
  - New-AzPrivateDNSZone, 108
  - New-AzTrafficManagerProfile, 139–140
  - NewAzVirtualNetwork, 12
  - New-AzVirtualNetworkGateway, 31–32, 52–53
  - New-AzVirtualNetworkGatewayConnection, 52–53
  - New-AzVirtualNetworkLink, 190–191, 199
  - creating a vNET, 12
  - DNS zones, creating, 108
  - linking a private DNS zone to a vNET, 111
  - private endpoints, creating, 190–191
  - Private Link service, creating, 199
  - reverse DNS lookups, creating, 118
  - setting up auto registration, 112
- Azure Private Link, 7, 183.
  - See *also* Private Link service
  - best practices, 200
  - DNS configuration, 186
  - features, 183–184
  - private endpoints, 184
    - features, 185
    - integration with Azure PaaS offerings, 184
    - integration with customer-owned services, 185
    - limitations, 185–186
    - security and, 186
- Azure Traffic Manager, 123
  - best practices, 141–144
  - endpoints, 124–125
    - Azure, 125
    - external, 125–126
    - failover and recovery, 134
    - monitoring, 133
    - nested, 126
  - features, 123–124
  - nested profiles, 126–127
  - RBAC and, 141
  - Real User Measurements, enabling, 141–143
  - setting up
    - using Azure CLI, 140–141
    - using Azure Portal, 134–139
- using Azure PowerShell, 139–140
- traffic routing, 127–128
  - geographic, 132
  - multi-value, 128
  - performance, 130–131
  - priority, 128–129
  - subnet-based, 128
  - weighted, 129–130
- Azure VPN gateway, 35
  - availability zone support, 79
  - best practices, 54
  - BGP and, 38
  - connection types, 37–38
  - creating
    - using Azure CLI, 53
    - using Azure Portal, 46–52
    - using Azure PowerShell, 52–53
  - deployment considerations, 36–37
  - design concepts, 36–37
  - ExpressRoute and, 45–46
  - features, 35–36
  - gateway SKUs, 37
  - gateway subnet, 38
  - local network gateways, 38
  - point-to-site VPN gateway
    - connections, 43
    - authentication, 43–44
    - Highly Available vNET-to-vNET, 45
    - supported protocols, 43
    - vNET-to-vNET
      - connections, 44
  - redundancy, 39
  - site-to-site VPN gateway
    - connections
      - active-active mode, 41–42
      - HA in active-active mode, 42
      - High Availability in active-standby mode, 41
      - multi-site active-Active mode, 40

- single-site active standby mode, 39–40
- zonal gateways, 46
- zone-redundant gateways, 46

AzureActiveDirectory service tag, 7

## B

- back ends, 147
- back-end pools, 1, 20, 57, 147
- basic listener, 21
- basic request routing rule, 21
- Basic SKU, 37
- best practices
  - application gateway, 33–34
  - Azure Bastion, 179–181
  - Azure DNS, 119–121
  - Azure Firewall, 100–102
  - Azure Front Door, 165–166
  - Azure Load Balancer, 74–75
  - Azure Private Link, 200
  - Azure Traffic Manager, 141–144
  - Azure VPN gateway, 54
  - for designing and securing vNETs, 12–14
- BGP (Border Gateway Protocol), 6, 38

## C

- caching, Azure Front Door, 151–153
- CDNs (content delivery networks), 145
- classic rules, 84
- commands
  - az network front-door create, 163–165

- az network lb create, 72–73
- az network private-dns link create, 111
- az network private-endpoint create, 191–192, 199
- az network traffic-manager profile create, 140–141
- az network vnet, 12
- Azure PowerShell
  - New-AzBastion, 175–176
  - New-AzDNSZone, 118
  - New-AzFirewall, 99–100
  - New-AzFrontDoor, 162–163
  - New-AzLoad Balancer, 71–72
  - New-AzPrivateDNSVirtualNetworkLink, 112
  - New-AzPrivateDNSZone, 108
  - New-AzTrafficManagerProfile, 139–140
  - New-AzVirtualNetworkGateway, 31–32, 52–53
  - New-AzVirtualNetworkGatewayConnection, 31–32
  - New-AzVirtualNetworkLink, 190–191, 199
- conditional access policies, 180
- connection types, VPN gateway, 37–38
- creating
  - alias record sets, 113–116
  - application gateway
    - using Azure CLI, 32–33
    - using Azure Portal, 24–30
    - using Azure PowerShell, 31–32
  - Azure Bastion
    - using Azure CLI, 175–176
    - using Azure Portal, 172–175

- using Azure PowerShell, 175–176
- Azure Firewall
  - using Azure Portal, 87–99
  - using Azure PowerShell, 99–100
- Azure Load Balancer
  - using Azure CLI, 72–73
  - using Azure Portal, 62–70
  - using Azure PowerShell, 71–72
- Azure VPN gateway
  - using Azure CLI, 53
  - using Azure Portal, 46–52
  - using Azure PowerShell, 52–53
- DNS zones
  - using Azure CLI, 108
  - using Azure Portal, 106–108
  - using Azure PowerShell, 108
- private endpoints
  - using Azure CLI, 191–192
  - using Azure Portal, 187–190
  - using Azure PowerShell, 190–191
- Private Link service
  - using Azure CLI, 199
  - using Azure Portal, 195–198
  - using Azure PowerShell, 199
- reverse DNS lookups, 116
  - using Azure CLI, 118
  - using Azure Portal, 116–118
  - using Azure PowerShell, 118
- vNET
  - using Azure CLI, 12
  - using Azure Portal, 8–11
  - using Azure PowerShell, 12
- custom routes
  - BGP routes, 6
  - user-defined routes, 5

**D**

DDoS (distributed denial of service) attacks, 13

default system routes, 3–4

deployment considerations

- Azure Load Balancer, 56
- Azure VPN gateway, 36–37
- vNET(s), 2–3

diagnostic logs, 87

disaster recovery, 8

- Azure Bastion, 171
- private endpoints and, 186–187

DNAT (destination NAT), Azure Firewall and, 79

DNS proxy, 86

DNS zones, 106, 186. *See also* private DNS zones; public DNS zones

- auto registration, 111–112
  - using Azure CLI, 112
  - using Azure Portal, 112
  - using Azure PowerShell, 112
- creating
  - using Azure CLI, 108
  - using Azure Portal, 106–108
  - using Azure PowerShell, 108
- linking to vNETs, 109
  - using Azure CLI, 111
  - using Azure Portal, 110
  - using Azure PowerShell, 111
- private, 105
- public, 105
- reverse DNS lookups, creating, 116
  - using Azure CLI, 118
  - using Azure Portal, 116–118
  - using Azure PowerShell, 118
- zone delegation, 119

dynamic content compression, 152

**E**

endpoints, 123, 124–125. *See also* private endpoints

- Azure, 125
- external, 125–126
- failover and recovery, 134
- monitoring, 133
- nested, 126
- private, 7, 184
  - disaster recovery and, 186–187
  - features, 185
  - integration with Azure PaaS offerings, 184
  - integration with customer-owned services, 185
  - limitations, 185–186
  - security and, 186
  - service, 7

ExpressRoute, 35, 45–46

external endpoints, 125–126

**F**

features

- Azure Application Gateway, 16–18
- Azure Bastion, 167–168
- Azure DNS, 103–104
- Azure Firewall, 78
- Azure Front Door, 145–146
- Azure Load Balancer, 56
- Azure Private Link, 183–184
- Azure Traffic Manager, 123–124
- Azure vNET (virtual network), 1–2
- Azure VPN gateway, 35–36
  - private endpoint, 185

file chunking, 152

filtering, threat intelligence-based, 83

forced tunneling, 83

FQDN tags, 81

front-end IP addresses, 19–20

- Azure Application Gateway, 19–20
- Azure Load Balancer, 57

**G**

gateway SKUs, 37

gateway subnet, 38

geo-filtering, 34, 165

geographic traffic routing, 132

global peering, 171

groupings, 81

- infrastructure FQDNs, 82
- IP groups, 82
- tags, 81
  - FQDN, 81
  - service, 82
- web categories, 82

**H**

hash-based algorithm, 60–61

health probes, 23

- Azure Application Gateway, 23
- Azure Front Door, 148
- Azure Load Balancer, 57–58

High Availability in active-standby mode, Azure VPN gateway, 41

high-availability ports load-balancing rule, 59

HTTP, 22

HTTP/2 protocols, 18

**I**

IKEv2 VPN, 43

inbound NAT rules, 59

infrastructure FQDNs, 82  
 internal load balancer, 55  
 IP addresses  
     Azure Bastion, 172  
     front-end, 19–20  
 IP groups, 82

## J-K-L

Key Vault, 180  
 latency-based traffic routing, 148  
 listeners, 20–21  
 load balancers, 15–16, 148.  
     *See also* Azure Application Gateway  
         algorithms, 60–61  
         internal, 55  
         public, 55  
         rules, 58  
         HA ports, 59  
         inbound NAT, 59  
         outbound SNAT, 60  
 local network gateways, 38  
 logging, 86  
     Activity logs, 87  
     Azure Front Door, 165  
     diagnostic logs, 87  
     firewall metrics, 86  
     network traffic, 166

## M

MFA (multi-factor authentication), 180  
 monitoring, endpoints, 133  
 multi-site active-Active mode, 40  
  
 multi-site listener, 21  
 multi-value traffic routing, 128

## N

NAT (network address translation), 7, 59  
 nested endpoints, 126  
 nested profiles, 126–127  
 network traffic filtering, 80  
 New-AzBastion command, 175–176  
 New-AzDNSZone command, 118  
 New-AzFirewall command, 99–100  
 New-AzFrontDoor command, 162–163  
 New-AzLoad Balancer command, 71–72  
 New-AzPrivateDNSV irtualNetworkLink command, 112  
 New-AzPrivateDNSZone command, 108  
 New-AzTrafficManagerProfile command, 139–140  
 NewAzVirtualNetwork command, 12  
 New-AzVirtualNetworkGateway command, 31–32, 52–53  
 New-AzVirtualNetworkGateway Connection command, 31–32  
 New-AzVirtualNetworkLink command, 190–191, 199  
 next-hop types, 3–5  
 non-zonal redundancy, 61  
 NSGs (network security groups), 6, 13

## O

OpenVPN protocol, 43  
 optional system routes, 4–5  
 outbound SNAT rules, 60

## P

path-based request routing rule, 21  
 path-based routing, 18  
 PAWs (privileged access workstations), 180  
 peering, 5  
     Azure Bastion, 171  
     vNET deployment and, 2–3  
 performance traffic routing, 130–131  
 permissions, Azure Bastion, 172  
 point-to-site VPN gateway connections  
     authentication, 43–44  
     Highly Available vNET-to-vNET, 45  
     vNET-to-vNET connections, 44  
 policy(ies)  
     Azure Firewall, 84  
     -based VPNs, 37  
     conditional access, 180  
     TLS, 23–24, 34  
 priority traffic routing, 128–129, 149  
 private DNS zones, 105. *See also* DNS zones  
     auto registration, 111–112  
         using Azure Portal, 112  
         using Azure PowerShell, 112  
     creating  
         using Azure CLI, 108  
         using Azure Portal, 106–108  
         using Azure PowerShell, 108  
     reverse DNS lookups, creating, 116  
         using Azure CLI, 118  
         using Azure Portal, 116–118  
         using Azure PowerShell, 118  
     private endpoints, 7, 184

- creating
  - using Azure CLI, 191–192
  - using Azure Portal, 187–190
  - using Azure PowerShell, 190–191
- disaster recovery and, 186–187
- features, 185
- integration
  - with Azure PaaS offerings, 184
  - with customer-owned services, 185
- limitations, 185–186
- security and, 186
- Private Link service, 192–194
  - access approvals, 195
  - connecting to, 194
  - creating
    - using Azure CLI, 199
    - using Azure Portal, 195–198
    - using Azure PowerShell, 199
  - High Availability, 195
  - limitations, 195
  - visibility options, 194–195
- profiles, nested, 126–127
- public DNS zones, 105, 113–116.
  - See also* alias record sets
- public load balancer, 55

## Q-R

- query
  - DNS, 119
  - strings, 152
- RBAC (role-based access control), 166
  - Azure DNS and, 119–120
  - Azure Traffic Manager and, 141
- Real User Measurements, 141–143
- redundancy

- Azure Bastion, 171
  - non-zonal, 62
  - VPN gateway, 39
  - zonal, 61
  - zone, 61
- regional peering, 171
- request routing rules
  - redirection support, 22
  - rewriting of HTTP headers and URLs, 22
- reverse DNS lookups, creating, 116
  - using Azure CLI, 118
  - using Azure Portal, 116–118
  - using Azure PowerShell, 118
- route-based VPNs, 37
- routing, 3
  - BGP routes, 6
  - default system routes, 3–4
  - NAT (network address translation), 7
  - next-hop types, 3–4
  - optional system routes, 4–5
  - path-based, 18
  - route selection, 6
  - user-defined routes, 5
- rules
  - Azure Firewall, order of processing, 84–85
  - classic, 84
  - load balancing, 58
    - HA ports, 59
  - inbound NAT, 59
  - outbound SNAT, 60
  - request routing, 21–22
    - redirection support, 22
    - rewriting of HTTP headers and URLs, 22
  - Threat Intelligence, 83
  - URL redirect, 149–150
  - URL rewrite, 149
  - WAF (web application firewall), 17
- Rules Engine, Azure Front Door, 151

## S

- scaling, 23, 171
- security
  - Azure Front Door
    - integration with Azure DDoS Protection Basic, 153
    - protection against unwanted protocols, 153
    - WAF (web application firewall), 154
  - private endpoints and, 186
  - vNET, 7
- segmentation, 7
- service endpoints, 7
- service tags, 7–8, 13, 82
- services, Private Link, 7, 192–194
  - access approvals, 195
  - connecting to a VM using Azure Portal, 194
  - High Availability, 195
  - limitations, 195
  - visibility options, 194–195
- session affinity, 149
- setting up Azure Traffic Manager
  - using Azure CLI, 140–141
  - using Azure Portal, 134–139
  - using Azure PowerShell, 139–140
- site-to-site VPN gateway
  - connections
    - active-active mode, 41–42
    - HA in active-active mode, 42
    - High Availability in active-standby mode, 41
    - multi-site active-Active mode, 40
    - single-site active standby mode, 39–40
- sizing, 34
- SKUs
  - Azure Bastion, 171
  - gateway, 37

- sizing and scaling, 23
- v2, 34
- SNAT (source NAT), Azure Firewall and, 79–80
- source IP affinity algorithm, 61
- SSTP (Secure Socket Tunneling Protocol (SSTP)), 43
- static VIP (virtual IP) assignment, 17
- subnet-based traffic routing, 128
- subnets, vNET deployment and, 2

## T

- tags, 81
  - FQDN, 81
  - service, 82
- Threat Intelligence, 83
- TLS policy, 23–24, 34
- traffic filtering, 80
  - application FQDN, 81
  - network, 80
  - web, 80
- traffic redirection
  - Azure Application Gateway, 17
  - Azure DNS and, 104
  - request routing rules and, 22
- traffic routing
  - Azure Front Door, 148–149
  - geographic, 132
  - latency-based, 148
  - multi-value, 128
  - performance, 130–131
  - priority, 128–129, 149
  - session affinity, 149
  - subnet-based, 128
  - weighted, 129–130, 149

## U

- URL redirect, 149–150
- URL rewrite, 149
- user-defined routes, 5

## V

- v2 SKU, 34
- virtual network service tags, 166
- VirtualNetwork service tag, 8
- visibility options, Private Link service, 194–195
- VMs (virtual machines), 1, 176–179
- VMSS (virtual machine scale sets), 1
- vNETs
  - availability zones, 6
  - best practices, 12–14
  - creating
    - using Azure CLI, 12
    - using Azure Portal, 8–11
    - using Azure PowerShell, 12
  - deployment considerations
    - address space, 2
    - peering, 2–3
    - subnets, 2
  - disaster recovery, 8
  - features, 1–2
  - gateway, 4
  - integrations for enhanced security, 7
  - linking to DNS zones, 109
    - using Azure CLI, 111
    - using Azure Portal, 110
    - using Azure PowerShell, 111
- NAT (network address translation), 7

- NSGs (network security groups), 6
- peering, 5
- routing, 3
  - BGP routes, 6
  - custom routes, 5
  - default system routes, 3–4
  - optional system routes, 4–5
  - user-defined routes, 5
- security, 7
- service tags, 7–8
- VPN (virtual private network), 1, 37

## W

- WAF (web application firewall), 16, 154
- web categories, 82
- web-traffic filtering, 80
- weighted traffic routing, 129–130, 149
- wildcard domains, 151

## X-Y-Z

- zonal
  - gateways, 46
  - redundancy, 61
- zone
  - delegation, 119
  - redundancy, 46, 61

# Proven best practices for success with every Azure networking service

For cloud environments to operate and scale optimally, their networking services must be designed, deployed, and managed well. Now, there's a complete, best-practice guide to doing just that. Writing for everyone involved in delivering Azure workloads and services, leading cloud consultant Avinash Valiramani provides a deep dive and practical field advice for Azure Virtual Networks, Azure VPN Gateways, Azure Load Balancing, Azure Traffic Manager, Azure Firewall, Azure DNS, Azure Bastion, Azure Front Door and more. Whatever your role in delivering efficient, scalable networking services, this guide will help you make the most of your Azure investment.

## Leading Azure consultant Avinash Valiramani shows how to:

- Use Azure Virtual Networks to establish a backbone for hosting other Azure resources
- Provide HTTP/HTTPS load-balancing and routing for web servers and apps through Azure Application Gateway
- Connect on-premises and other public networks to Azure for secure communications using the Azure VPN Gateway service
- Provide secure load balancing to apps from internal and public networks using Azure Load Balancer services
- Integrate Azure Firewall to centrally protect Azure resources across multiple subscriptions
- Access globally scaled, fully-managed DNS services with 100% SLA from the closest Azure DNS servers
- Provide optimal network routing to the closest application endpoint for public-facing applications with Azure Traffic Manager
- Use Microsoft's global edge network along with Azure Front Door to speed up access, harden security and enhance scalability for consuming-facing and internal web applications

## Also look for these Definitive Guides to Azure success:

- *Microsoft Azure Compute: The Definitive Guide*
- *Microsoft Azure Monitoring and Management: The Definitive Guide*
- *Microsoft Azure Storage: The Definitive Guide*

MicrosoftPressStore.com

ISBN-13: 978-0-13-756989-2  
ISBN-10: 0-13-756989-0



**U.S.A. \$44.99**

[Recommended]

Microsoft / Azure



## About this Book

- For IT, infrastructure, virtualization, and cloud admins and architects at all levels of Azure experience
- Especially useful to IT pros in midsized to large organizations who have deployed, operated, monitored, upgraded, migrated, or designed networking services

## About the Author

**Avinash Valiramani** is a Cloud Architect with more than 15 years of expertise in areas of Microsoft Technologies such as Microsoft Azure, Microsoft 365, Windows Server, Microsoft Exchange, SCCM, Intune, Hyper-V, and others. He is leading a successful cloud infrastructure and security consultancy primarily focusing on helping enterprises globally in their Cloud Roadmap Architecture and Onboarding/Migration/Security Strategies & Implementation Services. He has also authored a course on Azure Virtual Desktop for O'Reilly Media. Follow Avinash on Twitter at @avaliramani for updates on new events and books.

## Access code samples at:

MicrosoftPressStore.com/  
AzureNetworkingTDG/downloads

<https://github.com/avinashvaliramani/AzureNetworkingTDG>

Cover design by Chuti Prasertsith

Cover image by SEREE YINDEE/Shutterstock

Microsoft Press