Microsoft Azure Administrator

Exam Ref AZ-104

Harshul Patel

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Microsoft

Exam Ref AZ-104 Microsoft Azure Administrator

Harshul Patel

Exam Ref AZ-104 Microsoft Azure Administrator

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

Copyright © 2022 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

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: 978-013-680538-0
ISBN-10: 0-136-80538-8

Library of Congress Control Number: 2021936223

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 or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

CREDITS

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

SPONSORING EDITOR
Charvi Arora

DEVELOPMENT EDITOR
Rick Kughen

MANAGING EDITOR
Sandra Schroeder

PROJECT EDITOR
Tracey Croom

COPY EDITOR
Rick Kughen

INDEXER
Cheryl Ann Lenser

PROOFREADER
Donna E. Mulder

EDITORIAL ASSISTANT
Cindy Teeters

COMPOSITOR
codeMantra

COVER DESIGNER
Twist Creative, Seattle

Contents at a glance

This page intentionally left blank

Contents

This page intentionally left blank

Acknowledgments

I would like to acknowledge the flawless support I have received throughout the journey of book by Loretta and Charvi from the Pearson team. They have been very supportive and flexible, knowing the fact that I was dealing with multiple things at my end. I would also like to thank my wife, Divya, for her tremendous support in the making of this book. Despite her pregnancy, she played an instrumental role by encouraging and allowing me to complete the book on time. And last but not the least, the cuddle and cute smile of my little bundle of joy, Rivan, was a real energy booster during breaks in the middle of the night.

This page intentionally left blank

About the Author

HARSHUL PATEL is a technology enthusiast formerly from India who currently lives in Canada. He has been a cloud consultant with Microsoft Services for more than six year. He drives the adoption of Microsoft's cloud platforms for enterprise customers. He is thoroughly knowledgeable across various virtualization and cloud technologies. Harshul is an experienced author and an early adopter of many Microsoft products. He is a frequent speaker at various user group gatherings and a co-founder of a few global user groups.

Apart from work, Harshul is a happy-go-lucky guy. He loves to travel and spend time with his family and friends. Harshul and his wife, Divya, had a baby boy during the production of this book; they call him Rivan.

This page intentionally left blank

Introduction

The AZ-104 exam focuses on common tasks and concepts that an administrator needs to understand to deploy and manage infrastructure in Microsoft Azure. Manage Azure identities and Azure subscriptions is a key topic on the exam, which includes managing Azure AD objects (users, groups, and devices), use of Azure AD join and self-service password resets; it also covers role based access control, tagging, subscription level policies and resource organization using resource groups, subscription and management groups. Another topic covered is implement and manage storage, which includes creating and configuring storage accounts as well as configuring Azure files and understanding the services for importing and exporting data to Azure. A significant portion of the exam is focused on deploying and managing Azure compute resources, which includes configuring high availability of Azure VMs, creating and configuring virtual machine and their automated deployments as well as creating and configuring container solutions such as Azure Kubernetes Service (AKS) and Azure Container Instances (ACI); it also covers configuring web apps using app service and app service plans. This book also covers the creation and management of virtual networks, DNS, connectivity between virtual networks, configuring network security groups, Azure firewall and Azure bastion service; it also explains the load balancing solutions including configuration of application gateway. The final topic is monitor and backup Azure resources, which includes topics on how to monitor resources using Azure Monitor as well as how to implement back and recovery of Azure VMs including site to site recovery using Azure site recovery.

This book is geared toward Azure administrators who manage cloud services that span storage, security, networking and compute. It explains how to configure and deploy services across a broad range of related Azure services to help you prepare for the exam.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the reference links provided throughout this book and take the time to research and study the topic. Great information is available on Microsoft Docs.

Organization of this book

This book is organized by the "Skills measured" list published for the exam. The "Skills measured" list is available for each exam on the Microsoft Learning website: *https://aka.ms/examlist*. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in

each topic area determine a chapter's organization. If an exam covers six major topic areas, for example, the book will contain six chapters.

Preparing for the exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. This book is *not* designed to teach you new skills.

We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Exam Ref and another study guide for your "at home" preparation and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you. Learn more about available classroom training and find free online courses and live events at *http://microsoft.com/learn*. Microsoft Official Practice Tests are available for many exams at *http://aka.ms/practicetests*.

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

Microsoft certifications

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

MORE INFO **ALL MICROSOFT CERTIFICATIONS**

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

Quick access to online references

Throughout this book are addresses to webpages that the author has recommended you visit for more information. Some of these links can be very long and painstaking to type, so we've shortened them for you to make them easier to visit. We've also compiled them into a single list that readers of the print edition can refer to while they read.

Download the list at *MicrosoftPressStore.com/ExamRefAZ104/downloads*

The URLs are organized by chapter and heading. Every time you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

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/ExamRefAZ104/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*.

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*.

This page intentionally left blank

CHAPTER 2

Implement and manage storage

Implementing and managing storage is one of the most important aspects of building or deploying a new solution using Azure. There are several services and features available for use, and each has its own place. Azure Storage is the underlying storage for most of the services in Azure. It provides service for the storage and retrieval of files, and it has services that are available for storing large volumes of data through tables. Also, Azure Storage includes a fast and reliable messaging service for application developers with queues. In this chapter, we review how to implement and manage storage with an emphasis on Azure Storage.

Also, we discuss related services such as Import/Export, Azure Files, and many of the tools that simplify the management of these services.

Skills covered in this chapter:

- Skill 2.1: Secure Storage
- Skill 2.2: Manage Storage
- Skill 2.3: Configure Azure Files and Azure Blob Storage

Skill 2.1: Secure Storage

An Azure Storage account is an entity you create that is used to store Azure Storage data objects such as blobs, files, queues, tables, and disks. Data in an Azure Storage account is durable and highly available, secure, massively scalable, and accessible from anywhere in the world over HTTP or HTTPS.

This section covers how to:

- Configure network access to storage accounts
- Create and configure storage accounts
- Generate shared access signatures
- Manage access keys
- Configure Azure AD Authentication for a storage account

Configure network access to the storage accounts

Storage accounts are managed through Azure Resource Manager. Management operations are authenticated and authorized using Azure Active Directory and RBAC. Each storage account service exposes its own endpoint used to manage the data in that storage service (blobs in Blob Storage, entities in tables, and so on). These service-specific endpoints are not exposed through Azure Resource Manager; instead, they are (by default) Internet-facing endpoints.

Access to these Internet-facing storage endpoints must be secured, and Azure Storage provides several ways to do so. In this section, we will review the network-level access controls: the storage firewall and service endpoints. We also discuss Blob Storage access levels. The following sections then describe the application-level controls: shared access signatures and access keys. In later sections, we also discuss Azure Storage replication and how to leverage Azure AD authentication for a storage account.

Storage firewall

The storage firewall allows you to limit access to specific IP addresses or an IP address range. It applies to all storage account services (blobs, tables, queues, and files). For example, by limiting access to the IP address range of your company, access from other locations will be blocked. Service endpoints are used to restrict access to specific subnets within an Azure VNet.

To configure the storage firewall using the Azure portal, open the storage account blade and click **Firewalls And Virtual Networks**. Under **All Access From**, click **Selected Networks** to reveal the **Firewall** and **Virtual Network** settings, as shown in Figure 2-1.

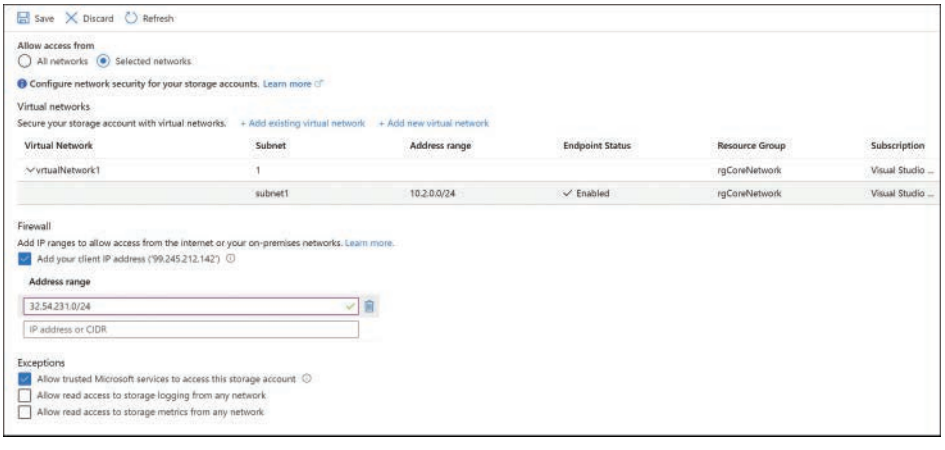

FIGURE 2-1 Configuring a storage account firewall and virtual network service endpoint access

When accessing the storage account via the Internet, use the storage firewall to specify the Internet-facing source IP addresses (for example, `32.54.231.0/24`, as shown in Figure 2-1) that will make the storage requests. All Internet traffic is denied, except the defined IP addresses

in the storage firewall. You can specify a list of either individual IPv4 addresses or IPv4 CIDR address ranges. (CIDR notation is explained in the chapter on Azure Networking.)

The storage firewall includes an option to allow access from trusted Microsoft services. These services include Azure Backup, Azure Site Recovery, and Azure Networking. For example, it will allow access to storage for NSG flow logs if the **Allow Trusted Microsoft Services To Access This Account** exceptions checkbox is selected (see Figure 2-1). It will also allow read-only access to storage metrics and logs.

> ***NOTE*** **ADDRESS SPACE FOR STORAGE FIREWALL**
>
> **When creating a storage firewall, you must use public Internet IP address space. You cannot use IPs in the private IP address space.**

Virtual network service endpoints

In some scenarios, a storage account is only accessed from within an Azure virtual network. In this case, it is desirable from a security standpoint to block all Internet access. Configuring virtual network service endpoints for your Azure Storage accounts allows you to remove access from the public Internet and only allow traffic from a virtual network for improved security.

Another benefit of using service endpoints is optimized routing. Service endpoints create a direct network route from the virtual network to the storage service. If forced tunneling is being used to force Internet traffic to your on-premises network or to another network appliance, requests to Azure Storage will follow that same route. By using service endpoints, you can use direct route to the storage account instead of the on-premises route, so no additional latency is incurred.

Configuring service endpoints requires two steps. First, from the virtual network subnet, choose **Microsoft.Storage** from the **Service Endpoints** drop-down menu. This creates the route from the subnet to the storage service but does not restrict which storage account the virtual network can use. To update the subnet settings, you should choose **virtualNetwork1** from the **Virtual Networks** blade. Then go to **Subnets** in the left pane under **Settings**. Click **Subnet1** to access the subnet settings. Figure 2-2 shows the subnet settings, including the service endpoint configuration.

The second step is to configure which virtual networks can access a particular storage account. From the storage account blade, click **Firewalls And Virtual Networks**. Under **All Access From**, click **Selected Networks** to reveal the **Firewall** and **Virtual Network** settings, as shown previously in Figure 2-1. Under **Virtual Networks**, select the virtual networks and subnets that should have access to this storage account.

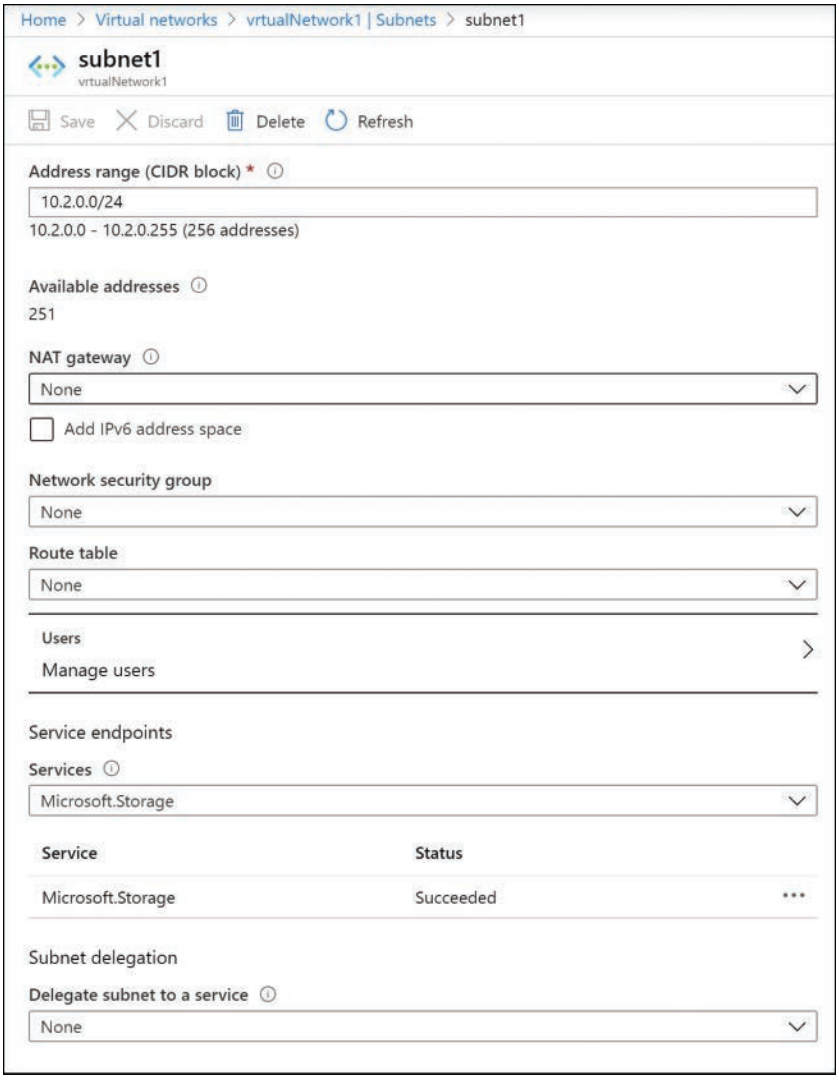

FIGURE 2-2 Configuring a subnet with a service endpoint for Azure Storage

Blob Storage access levels

Storage accounts support an additional access control mechanism that is limited only to Blob Storage. By default, no public read access is enabled for anonymous users, and only users with rights granted through RBAC or with the storage account name and key will have access to the stored blobs. To enable anonymous user access, you must change the container access level (see Figure 2-3). The supported levels are as follows:

- **Private.** With this option, only the storage account owner can access the container and its blobs. No one else would have access to them.

- **Blob.** With this option, only blobs within the container can be accessed anonymously.
- **Container.** With this option, blobs and their containers can be accessed anonymously.

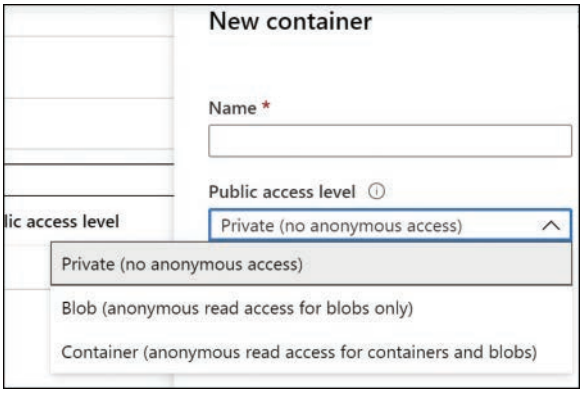

FIGURE 2-3 Blob Storage access levels

You can change the access level through the Azure portal, Azure PowerShell, Azure CLI, programmatically using the REST API, or by using Azure Storage Explorer. The access level is configured separately on each blob container.

A shared access signature token (SAS token) is a URI query string parameter that grants access to specific containers, blobs, queues, and tables. Use an SAS token to grant access to a client that should not have access to the entire contents of the storage account (and therefore, should not have access to the storage account keys) but still requires secure authentication. By distributing an SAS URI to these clients, you can grant them access to a specific resource, for a specified period of time, and with a specified set of permissions. Frequently, SAS tokens are used to read and write the data to users' storage accounts. Also, SAS tokens are widely used to copy blobs or files to another storage account.

> ***NOTE*** **SAS TOKENS USING HTTPS**
>
> **When dealing with SAS tokens, you must use only the HTTPS protocol. Because active SAS tokens provide direct authentication to your storage account, you must use a secure connection, such as HTTPS, to distribute SAS token URIs.**

Create and configure storage accounts

Azure Storage accounts provide a cloud-based storage service that is highly scalable, available, performant, and durable. Within each storage account, a number of separate storage services are provided:

- **Blobs.** Provides a highly scalable service for storing arbitrary data objects such as text or binary data.

- **Tables.** Provides a NoSQL-style store for storing structured data. Unlike a relational database, tables in Azure storage do not require a fixed schema, so different entries in the same table can have different fields.
- **Queues.** Provides reliable message queueing between application components.
- **Files.** Provides managed file shares that can be used by Azure VMs or on-premises servers.
- **Disks.** Provides a persistent storage volume for Azure VM which can be attached as a virtual hard disk.

There are three types of storage blobs: Block Blobs, Append Blobs, and Page Blobs. Page Blobs are generally used to store VHD files when deploying unmanaged disks. (Unmanaged disks are an older disk storage technology for Azure virtual machines. Managed disks are recommended for new deployments.)

When creating a storage account, there are several options that must be set: Performance Tier, Account Kind, Replication Option, and Access Tier. There are some interactions between these settings. For example, only the Standard performance tier allows you to choose the access tier. The following sections describe each of these settings. We then describe how to create storage accounts using the Azure portal, PowerShell, and Azure CLI.

Naming storage accounts

While naming an Azure Storage Account, you need to remember these points:

- The storage account name must be unique across all existing storage account names in Azure.
- The name must be between 3 to 24 characters and can contain only lowercase letters and numbers.

Performance tiers

When creating a storage account, you must choose between the Standard and Premium performance tiers. This setting cannot be changed later.

- **Standard.** This tier supports all storage services: blobs, tables, files, queues, and unmanaged Azure virtual machine disks. It uses magnetic disks to provide cost-efficient and reliable storage.
- **Premium.** This tier is designed to support workloads with greater demands on I/O and is backed by high-performance SSD disks. It only supports General-Purpose accounts with Disk Blobs and Page Blobs. It also supports Block Blobs or Append Blobs with BlockBlobStorage accounts and files with FileStorage accounts.

NOTE **REPLICATION OPTIONS WITH PREMIUM TIER**

Premium tier only supports LRS as a replication option for general-purpose storage accounts. It supports LRS and ZRS, both for BlockBlobStorage and FileStorage accounts.

Account kind

There are three possible values for the Standard tier: StorageV2 (General-Purpose V2), Storage (General-Purpose V1), and BlobStorage. There are four possible values for the Premium tier: StorageV2 (General-Purpose V2), Storage (General-Purpose V1), BlockBlobStorage, and FileStorage. Table 2-1 shows the features for each kind of account. Key points to remember are as follows:

- The Blob Storage account is a specialized storage account used to store Block Blobs and Append Blobs. You can't store Page Blobs in these accounts; therefore, you can't use them for unmanaged disks.
- Only General-Purpose V2 and Blob Storage accounts support the Hot, Cool, and Archive access tiers.

General-Purpose V1 and Blob Storage accounts can both be upgraded to a General-Purpose V2 account. This operation is irreversible. No other changes to the account kind are supported.

TABLE 2-1 Storage account types and their supported features

	General-Purpose V2	General-Purpose V1	Blob Storage	Block Blob Storage	File Storage
Services supported	Blob, File, Queue, Table	Blob, File, Queue, Table	Blob (Block Blobs and Append Blobs only)	Blob (Block Blobs and Append Blobs only)	File only
Unmanaged DIsk (Page Blob) support	Yes	Yes	No	No	No
Supported Performance Tiers	Standard, Premium	Standard, Premium	Standard	Premium	Premium
Supported Access Tiers	Hot, Cool, Archive	N/A	Hot, Cool, Archive	N/A	N/A
Replication Options	LRS, ZRS, GRS, RA-GRS, GZRS, RA-GZRS	LRS, GRS, RA-GRS	LRS, GRS, RA-GRS	LRS, ZRS	LRS, ZRS

Replication options

When you create a storage account, you can also specify how your data will be replicated for redundancy and resistance to failure. There are four options, as described in Table 2-2.

TABLE 2-2 Storage account replication options

Replication Type	Description
Locally redundant storage (LRS)	Makes three synchronous copies of your data within a single datacenter. Available for General-Purpose or Blob Storage accounts at both the Standard and Premium Performance tiers.

Replication Type	Description
Zone redundant storage (ZRS)	Makes three synchronous copies to three separate availability zones within a single region. Available for General-Purpose V2 storage accounts only, at the Standard Performance tier only. Also available for BlockBlobStorage and FileStorage.
Geographically redundant storage (GRS)	This is the same as LRS (three local copies), plus three additional asynchronous copies to a second datacenter hundreds of miles away from the primary region. Data replication typically occurs within 15 minutes, although no SLA is provided. Available for General-Purpose or Blob Storage accounts, at the Standard Performance tier only.
Read access geographically redundant storage (RA-GRS)	This has the same capabilities as GRS, plus you have read-only access to the data in the secondary datacenter. Available for General-Purpose or Blob Storage accounts, at the Standard Performance tier only.
Geographically zone redundant storage (GZRS)	This is the same as ZRS (three synchronous copies across multiple availability zones), plus three additional asynchronous copies to a second datacenter hundreds of miles away from the primary region. Data replication typically occurs within 15 minutes, although no SLA is provided. Available for General-Purpose v2 storage accounts only, at the Standard Performance tier only.
Read access geographically zone redundant storage (RA-GZRS)	This has the same capabilities as GZRS, plus you have read-only access to the data in the secondary datacenter. Available for General-Purpose V2 storage accounts only at the Standard Performance tier only.

NOTE **REPLICATION OPTIONS**

These replication options control the level of durability and availability of the storage account. When the entire datacenter is unavailable, LRS would incur an outage. If the primary region is unavailable, both the LRS and ZRS options would incur an outage, but the GRS and GZRS options would still provide the secondary region that takes care of the requests during the outage. However, not all the replication options are available in all regions. You can find supported regions with these replication options at *https://docs.microsoft.com/azure/storage/common/storage-redundancy*.

NOTE **SPECIFYING REPLICATION AND PERFORMANCE TIER SETTINGS**

When creating a storage account via the Azure portal, the replication and performance tier options are specified using separate settings. When creating an account using Azure PowerShell, the Azure CLI, or via a template, these settings are combined within the SKU setting.

For example, to specify a Standard storage account using locally redundant storage using the Azure CLI, use `--sku Standard_LRS`.

Access tiers

Azure Blob Storage supports three access tiers: Hot, Cool, and Archive. Each represents a trade-off of performance, availability, and cost. There is no trade-off on the durability (probability of data loss), which is extremely high across all tiers.

> ***NOTE*** **BLOB STORAGE ONLY**
>
> **Access tiers apply to Blob Storage only. They do not apply to other storage services, including Block Blob Storage.**

The tiers are as follows:

- **Hot.** This access tier is used to store frequently accessed objects. Relative to other tiers, data access costs are low while storage costs are higher.
- **Cool.** This access tier is used to store large amounts of data that is not accessed frequently and that is stored for at least 30 days. The availability SLA is lower than for the Hot tier. Relative to the Hot tier, data access costs are higher and storage costs are lower.
- **Archive.** This access tier is used to archive data for long-term storage, that is accessed rarely, can tolerate several hours of retrieval latency, and will remain in the Archive tier for at least 180 days. This tier is the most cost-effective option for storing data, but accessing that data is more expensive than accessing data in the Hot or Cool tiers.

New blobs will default to the access tier that is set at the storage account level, though you can override that at the blob level by setting a different access tier, including the archive tier.

> ***NOTE*** **ARCHIVE TIER SUPPORTABILITY**
>
> **Currently, the archive tier is not supported for ZRS, GZRS, or RA-GZRS accounts.**

Creating an Azure Storage account

To create a storage account by using the Azure portal, first click **Create A Resource** and then select **Storage**. Next, click **Storage Account**, which will open the **Create Storage Account** blade (see Figure 2-4). You must choose a unique name for the storage account name. Storage account names must be globally unique and may only contain lowercase characters and digits. Select the Azure region (Location), the performance tier, the kind of storage account, the replication mode, and the access tier. The blade adjusts based on the settings you choose so that you cannot select an unsupported feature combination.

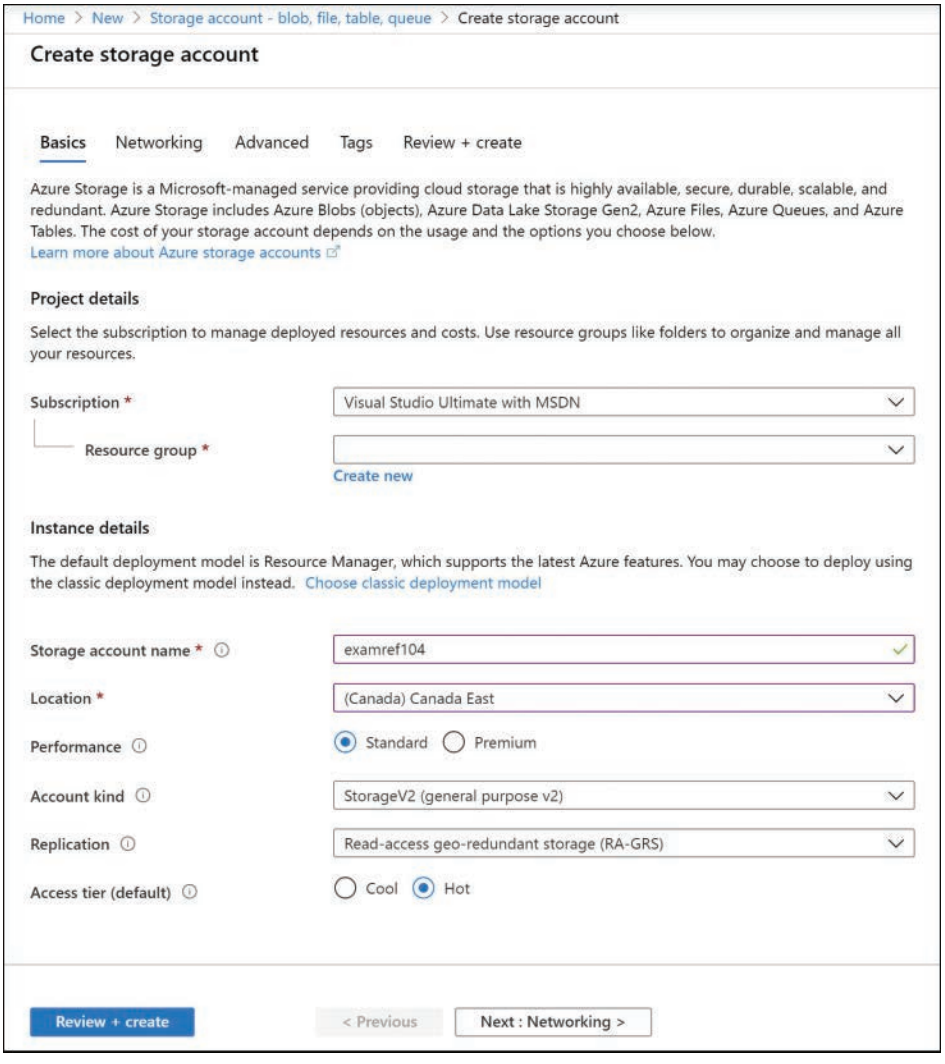

FIGURE 2-4 Creating an Azure storage account using the Azure portal

The **Networking** tab of the **Create Storage Account** blade is shown in Figure 2-5. This tab allows us to maintain storage account access either publicly by choosing **Public Endpoint (Selected Networks)** or privately by choosing **Private Endpoint**.

The **Advanced** tab of the **Create Storage Account** blade is shown in Figure 2-6. This tab allows you to specify whether SSL is required for accessing objects in storage; disabling or enabling Azure Files support; choosing data protection options such as blob Soft Delete or

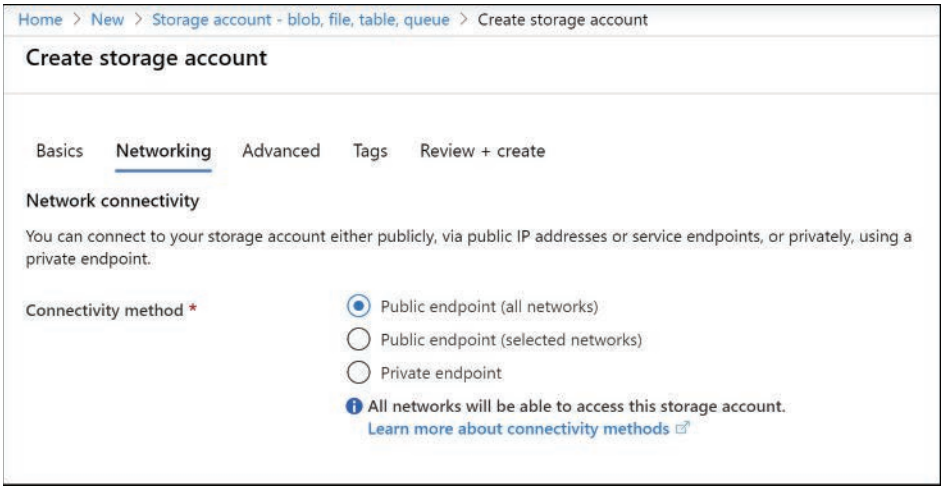

FIGURE 2-5 The networking properties that can be set when creating an Azure Storage account using the portal

Versioning; and for enabling Data Lake Storage integration. Additionally, clicking the **Tags** tab allows you to specify tags on the storage account resource.

> ***MORE INFO*** **CREATING A STORAGE ACCOUNT WITH POWERSHELL**
>
> **You can learn more about the additional parameters at *https://docs.microsoft.com/en-us/powershell/module/az.storage/new-azstorageaccount*.**

> ***MORE INFO*** **CREATING A STORAGE ACCOUNT WITH THE AZURE CLI**
>
> **You can learn more about the additional parameters at *https://docs.microsoft.com/cli/azure/storage/account#az-storage-account-create*.**

Generate shared access signatures

There are few different ways you can create an SAS token. An SAS token is a way to granularly control how a client can access data in Azure storage account. You can also use an account-level SAS to access the account itself. You can control many things, such as what services and resources the client can access, what permission the client has, how long the token is valid for, and more.

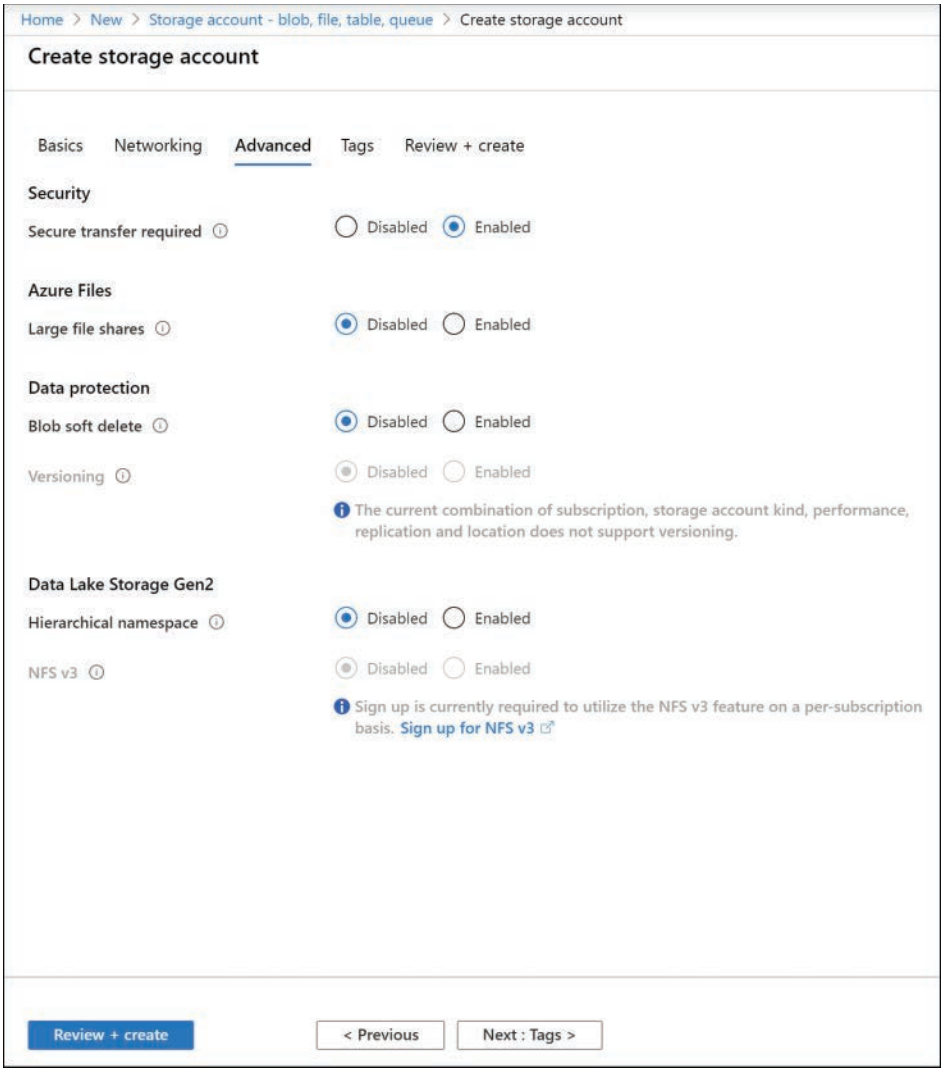

FIGURE 2-6 The advanced properties that can be set when creating an Azure Storage account using the Azure portal

In this section, we examine how to create SAS tokens using various methods. The simplest way to create one is by using the Azure portal. Browse to an Azure storage account and open the **Shared Access Signature** blade (see Figure 2-7). You can check the services, resource types, and permissions based on specific requirements, along with the duration for the SAS token validity and the IP addresses that are providing access. Lastly, you have an option to choose which key you want to use as the signing key for this token.

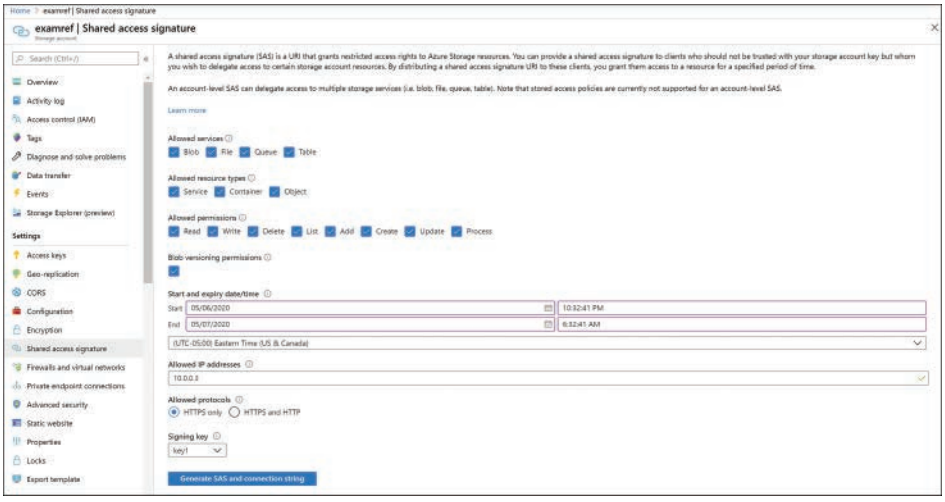

FIGURE 2-7 Creating a shared access signature using the Azure portal

Once the token is generated, it will be listed along with connection string and SAS URLs, as shown in Figure 2-8.

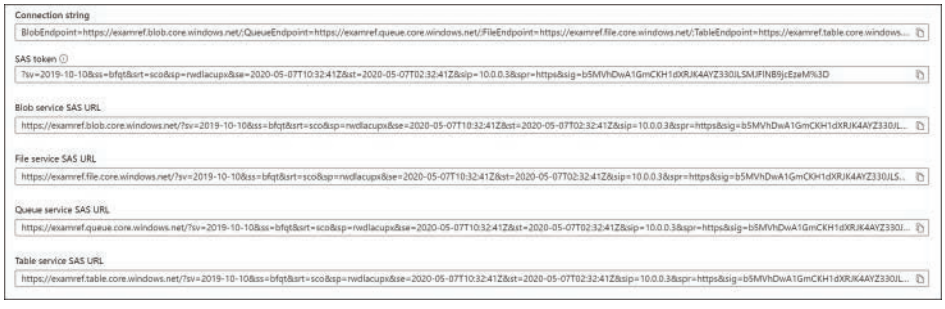

FIGURE 2-8 Generated SAS token with connection string and SAS URLs

Also, you can create SAS tokens using Storage Explorer or the command-line tools (or programmatically using the REST APIs/SDK). To create an SAS token using Storage Explorer, you need to first select the resource (storage account, container, blob, and so on) for which the SAS token needs to be created. Then right-click the resource and select **Get Shared Access Signature**. Figure 2-9 demonstrates how to create an SAS token using Azure Storage Explorer.

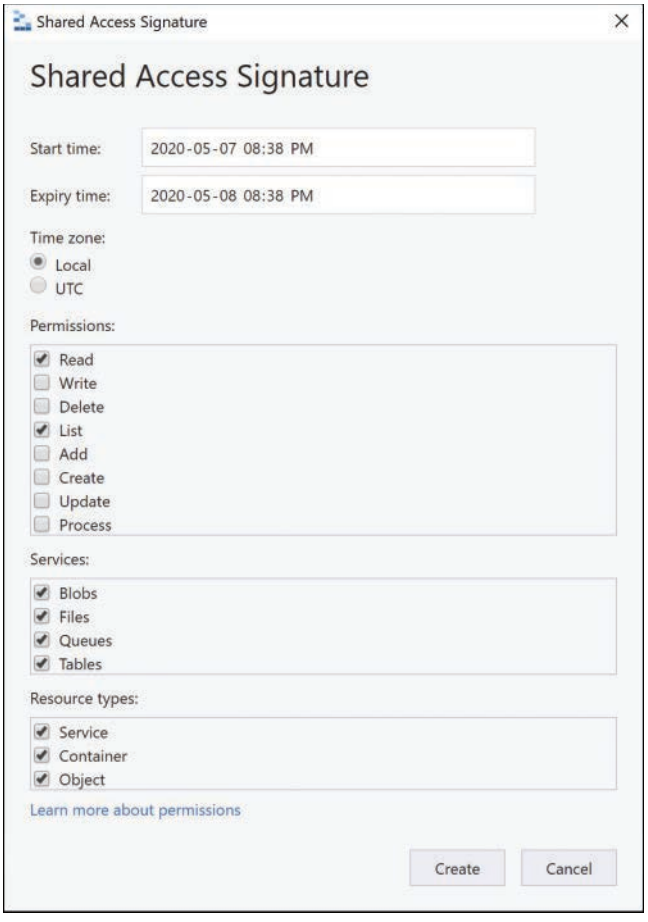

FIGURE 2-9 Creating a shared access signature using Azure Storage Explorer

Using shared access signatures

Each SAS token is a query string parameter that can be appended to the full URI of the blob or other storage resource for which the SAS token was created. Create the SAS URI by appending the SAS token to the full URI of the blob or other storage resource.

The following example shows the combination in more detail. Suppose the storage account name is `examref`, the blob container name is `examrefcontainer`, and the blob path is `sample-file.png`. The full URI to the blob in storage is

```
https://examrefstorage.blob.core.windows.net/examrefcontainer/sample-file.png
```

The combined URI with the generated SAS token is

```
https://examrefstorage.blob.core.windows.net/examrefcontainer/sample-file.png?
sv=2019-10-10&ss=bfqt&srt=sco&sp=rwdlacupx&se=2020-05-08T08:50:14Z&st=2020-05-08T00:
50:14Z&spr=https&sig=65tNhZtj2lu0tih8HQtK7aEL9YCIpGGprZocXjiQ%2Fko%3D
```

Using account-level SAS

You can create the SAS at the storage account–level, too. With this SAS, you can manage all the resources belonging to the storage account. You can also perform write and delete operations for all the resources (blobs, tables, and so on) of the storage account.

Currently, stored access policy is not supported for account-level SAS.

MORE INFO **ACCOUNT LEVEL SAS**

You can learn more about the account level SAS here: *https://docs.microsoft.com/rest/api/storageservices/create-account-sas*.

Using user delegation SAS

You can also create user delegation SAS using Azure AD credentials. The user delegation SAS is only supported by the Blob Storage, and it can grant access to containers and blobs. Currently, SAS is not supported for user delegation SAS.

MORE INFO **USER DELEGATION SAS**

You can learn more about the user delegation SAS at *https://docs.microsoft.com/rest/api/storageservices/create-user-delegation-sas*.

Using a stored access policy

An SAS token incorporates the access parameters (start and end time, permissions, and so on) as part of the token. The parameters cannot be changed without generating a new token, and the only way to revoke an existing token before its expiry time is to roll over the storage account key used to generate the token or delete the blob. In practice, these limitations can make standard SAS tokens difficult to manage.

Stored access policies allow the parameters for an SAS token to be decoupled from the token itself. The access policy specifies the start time, end time, and access permissions, and the access policy is created independently of the SAS tokens. SAS tokens are generated that reference the stored access policy instead of embedding the access parameters explicitly.

With this arrangement, the parameters of existing tokens can be modified by simply editing the stored access policy. Existing SAS tokens remain valid and use the updated parameters. You can revoke the SAS token by deleting the access policy, renaming it (changing the identifier), or changing the expiry time.

MORE INFO **STORED ACCESS POLICY EFFECT**

It can take up to 30 seconds for a stored access policy to take effect, and users might see an HTTP 403 when attempting access during that time.

Figure 2-10 shows the creation of stored access policies in the Azure portal.

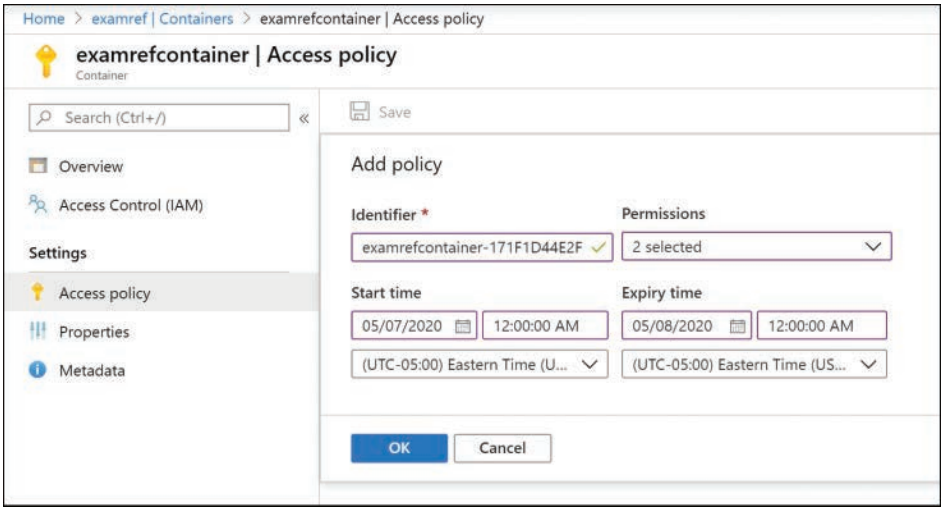

FIGURE 2-10 Creating stored access policies using Azure portal

Figure 2-11 shows stored access policies being created in Azure Storage Explorer.

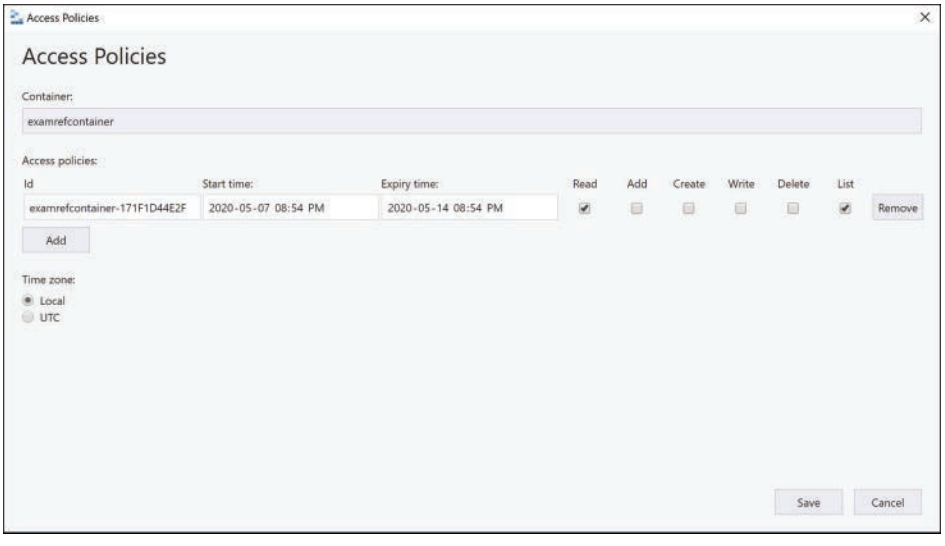

FIGURE 2-11 Creating stored access policies using Azure Storage Explorer

To use the created policies, reference them by name when creating an SAS token using Storage Explorer or when creating an SAS token using PowerShell or the CLI tools.

MORE INFO **MAX ACCESS POLICIES**

You can only have a max of five access policies on a container, table, queue, or file share.

Manage access keys

The simplest way to manage access to a storage account is to use access keys. With the storage account name and an access key of the Azure storage account, you have full access to all data in all services within the storage account. You can create, read, update, and delete containers, blobs, tables, queues, and file shares. In addition, you have full administrative access to everything other than the storage account itself. (You cannot delete the storage account or change settings on the storage account, such as its type.)

Applications will use the storage account name and key for access to Azure Storage. Sometimes, this is to grant access by generating an SAS token, and sometimes, it is for direct access with the name and key.

To access the storage account name and key, open the storage account from within the Azure portal and click **Access Keys**. Figure 2-12 shows the primary and secondary access keys for the `examref` storage account.

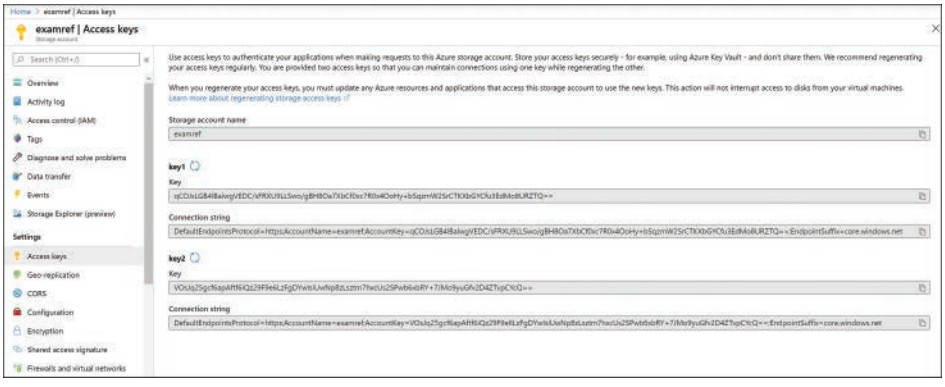

FIGURE 2-12 Access keys for an Azure storage account

Each storage account has two access keys. This allows you to modify applications to use the second key instead of the first and then regenerate the first key. This technique is known as "key rolling," and it allows you to reset the primary key with no downtime for applications that directly access storage using an access key.

Storage account access keys can be regenerated using the Azure portal or the command-line tools. In PowerShell, this is accomplished with the `New-AzStorageAccountKey` cmdlet; with Azure CLI, you will use the `az storage account keys renew` command.

NOTE **ACCESS KEYS AND SAS TOKENS**

Rolling a storage account access key will invalidate any SAS tokens that were generated using that key.

Managing access keys in Azure Key Vault

It is important to protect the storage account access keys because they provide full access to the storage account. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services, such as authentication keys, storage account keys, data encryption keys, and certificate private keys.

Keys in Azure Key Vault can be protected in software or by using hardware security modules (HSMs). HSM keys can be generated in place or imported. Importing keys is often referred to as bring your own key, or BYOK.

MORE INFO **USING HSM-PROTECTED KEYS FOR AZURE KEY VAULT**

You can learn more about the bring your own key (BYOK) scenario here: *https://docs.microsoft.com/azure/key-vault/key-vault-hsm-protected-keys*.

You can manage storage account keys with key vault using Azure PowerShell or CLI. You can learn more using the following links:

- **PowerShell: *https://docs.microsoft.com/azure/key-vault/secrets/overview-storage-keys-powershell***
- **CLI: *https://docs.microsoft.com/azure/key-vault/secrets/overview-storage-keys***

Accessing and unencrypting the stored keys is typically done by a developer, although keys from Key Vault can also be accessed from ARM templates during deployment.

MORE INFO **ACCESSING ENCRYPTED KEYS FROM AZURE KEY VAULT**

You can learn more about how developers securely retrieve and use secrets from Azure Key Vault here: *https://docs.microsoft.com/azure/storage/blobs/storage-encrypt-decrypt-blobs-key-vault*.

Configure Azure AD Authentication for a storage account

Azure AD authentication is beneficial for large customers who want to control the data access at an enterprise level based on their security and compliance standards. AAD authentication was recently added to the list in addition to existing shared-key and SAS token authorization mechanisms for Azure Storage (Blob and Queue). Azure blobs and queues are supported by Azure AD authentication. Azure Table storage is not supported with Azure AD authorization as of now.

NOTE **AZURE AD AUTHORIZATION SUPPORT FOR STORAGE ACCOUNTS**

Storage accounts that are created with the Azure Resource Manager deployment model only support Azure AD authorization.

AAD authentication enables customers to leverage Azure's RBAC for granting the required permissions to a security principal (users, groups, and applications) down to the scope of an individual blob container or queue. While authenticating a request, Azure AD returns an OAuth 2.0 token to security principal, which can be used for authorization against Azure Storage (blob or queue).

Azure AD authorization can be implemented in many ways, such as assigning a RBAC roles to a security principal (users, groups, and applications), using a managed service identity (MSI), or creating shared access signatures signed by Azure AD credentials and so on.

If an application is running from within an Azure entity such as an Azure VM, a virtual machine scale set, or an Azure Functions app, it can use a managed service identity (MSI) to access blobs or queues.

NEED MORE REVIEW? **AUTHORIZING ACCESS**

More information about authorizing access to blob and queue data with managed identities for Azure resources can be found at *https://docs.microsoft.com/en-us/azure/storage/common/storage-auth-aad-msi*

RBAC roles for blobs and queues

There are few built-in RBAC roles available in Azure for authorizing access to Blob and Queue Storage.

- **Storage Blob Data Owner:** Sets ownership and manages POSIX access control for Azure Data Lake Storage Gen2.
- **Storage Blob Data Contributor:** Grants read/write/delete permissions for Blob Storage.
- **Storage Blob Data Reader:** Grants read-only permissions for Blob Storage.
- **Storage Queue Data Contributor:** Grants read/write/delete permissions for Queue Storage.
- **Storage Queue Data Reader:** Grants read-only permissions for Queue Storage.
- **Storage Queue Data Message Processor:** Grants peek, retrieve, and delete permissions to messages in queues.
- **Storage Queue Data Message Sender:** Grants add permissions to messages in queues.

NEED MORE REVIEW? **BUILT-IN ROLE DETAILS**

For more information about built-in roles, see *https://docs.microsoft.com/azure/role-based-access-control/built-in-roles#storage*.

Resource scope for blobs and queues

It is also important to determine the scope of the access for security principal before you assign an RBAC role. You can narrow down the scope to the container or queue level. Below are the valid scopes:

- **Container.** Under this scope, the role assignment will be applicable at the container level. All the blobs inside the container, the container properties, and the metadata will inherit the role assignment when this scope is selected.
- **Queue.** Under this scope, the role assignment will be applicable at the queue level. All the messages inside the queue, as well as queue properties and metadata will inherit the role assignment when this scope is selected.
- **Storage account.** Under this scope, the role assignment will be applicable at the storage account level. All the containers, blobs, queues, and messages within the storage account will inherit the role assignment when this scope is selected.
- **Resource group.** Under this scope, the role assignment will be applicable at the resource group level. All the containers or queues in all the storage accounts in the resource group will inherit the role assignment when this scope is selected.
- **Subscription.** Under this scope, the role assignment will be applicable at the subscription level. All the containers or queues in all the storage accounts in all the resource groups in the subscription will inherit the role assignment when this scope is selected.

AAD authentication and authorization in Azure portal

In the following example, you will learn how to configure the AAD authentication method in order to allow users to access the blob data.

In Figure 2-13, you can see the `examrefcontainer` container has one blob named `UserCreateTemplate.csv`. Also, notice that the authentication method is currently set as **Access Key**.

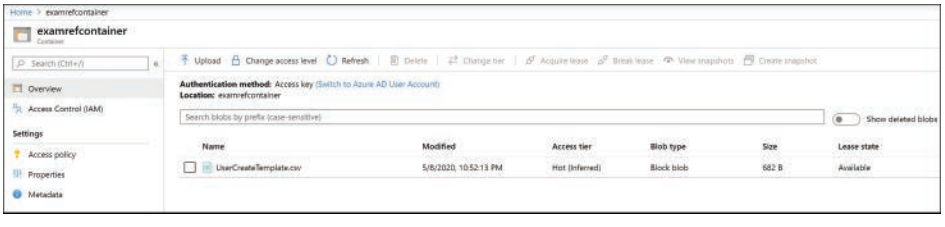

FIGURE 2-13 The overview blade of examrefcontainer

Switch the authentication method to **Azure AD User Account** by clicking **Switch To Azure AD Account**. You will see a warning message indicating that you do not have permission to list the data (see Figure 2-14).

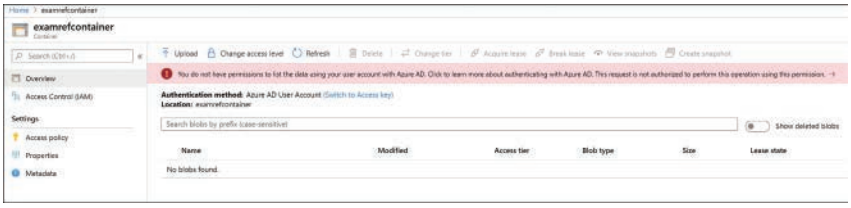

FIGURE 2-14 The overview blade of `examrefcontainer`

Now let's assign **Storage Blob Data Reader** role to the logged in user at container level. Go to the **Access Control (IAM)** blade on the container and select **Role** from the **Storage Blob Data Reader** drop-down menu. Then search for and select **CIE Administrator**. Click **Save** to apply the role assignment (see Figure 2-15).

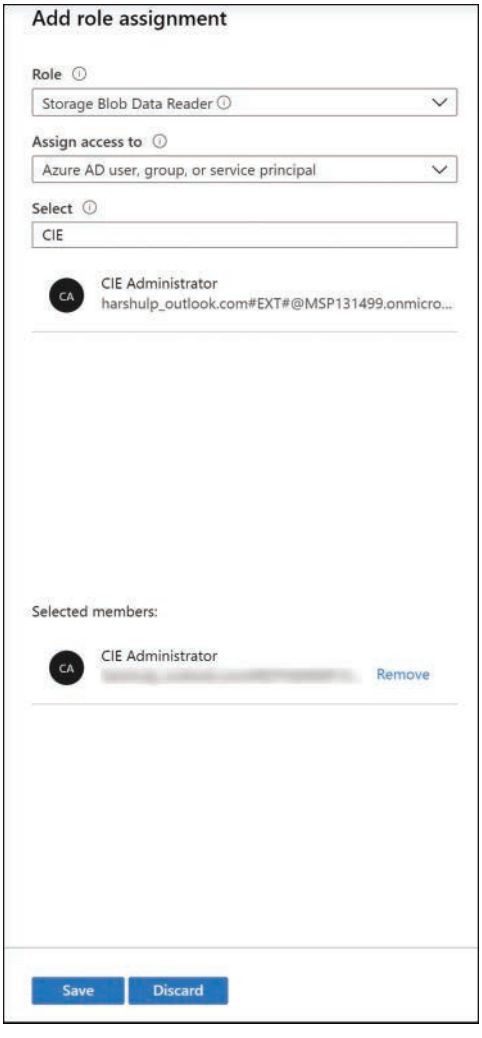

FIGURE 2-15 Storage Blob Data Reader Role assignment

You should now see the current user with the role **Storage Blob Data Reader**, which appears under **Role Assignments** (see Figure 2-16).

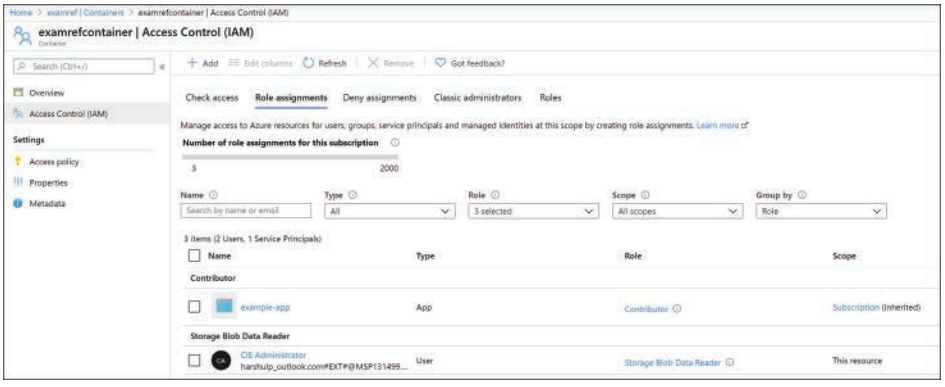

FIGURE 2-16 Role assignments for examrefcontainer

If you navigate to **Overview** blade of `examrefcontainer` now, you will see the `UserCreateTemplate.csv` blob with authentication method shown as **Azure AD User Account** (see Figure 2-17).

> ***NOTE*** **RBAC ROLES EFFECT**
>
> **Sometimes, RBAC roles take up to 5 minutes to propagate the role assignments.**

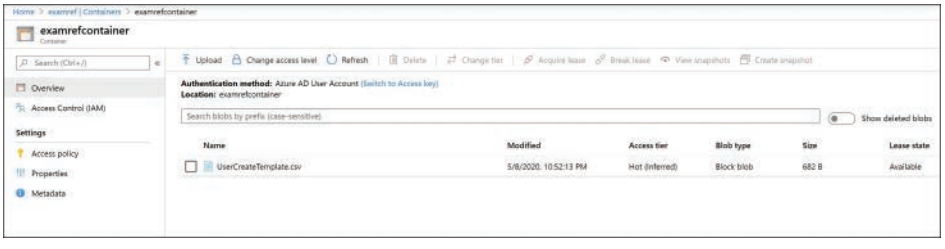

FIGURE 2-17 The overview blade of examrefcontainer

Configure access to Azure Files

Azure Files provides managed file shares that are accessible over the SMB protocol. SMB is a network file-sharing protocol, and Azure Files provides flexibility to use the following two types of identity-based authentication to access the shares.

- On-premises Active Directory Domain Services (AD DS)
- Azure Active Directory Domain Services (Azure AD DS)

In this section, you will learn how to use either of these domain services to access file shares over SMB. Azure file shares leverage Kerberos tokens to authenticate a user or application to access the file shares. You can configure authorization either at the share or directory/file levels.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z