

Cloud
Computing

Caltech

Center for Technology &
Management Education

Post Graduate Program in Cloud Computing

Cloud Computing

Caltech

**Center for Technology &
Management Education**

**PG CC - Microsoft Azure Architect
Design: AZ:304**



Select an Appropriate Storage Account

Learning Objectives

By the end of this lesson, you will be able to:

- 🕒 Analyze the Azure Storage Account
- 🕒 Illustrate Azure Storage services
- 🕒 Design Blob Performance Tiers
- 🕒 Configure uploading Blobs



Learning Objectives

By the end of this lesson, you will be able to:

- Design Access levels for Blob Containers
- Illustrate RBAC Roles for Access
- Design Azure Storage Explorer
- Configure and manage Tiered Storage using Azure Tools



A Day in the Life of an Azure Architect

You are working as an architect in an organization and you have been asked to decide on right choice of Azure storage for the below use cases:

- An application that will be used to store the e-documents as a part of digital transformation journey in your organization.
- Ensure that the overall cost of data storage is minimized and documents are stored in such a way that depending on the document age they move to the right tier of the storage.
- Take care of the storage access keeping the data security in mind by providing some authentication method to access the data.
- **A solution for storing vast volumes of unstructured data.**

To achieve all of the above along with some additional features, we would be learning a few concepts in this lesson that will help you find a solution for the above scenario.

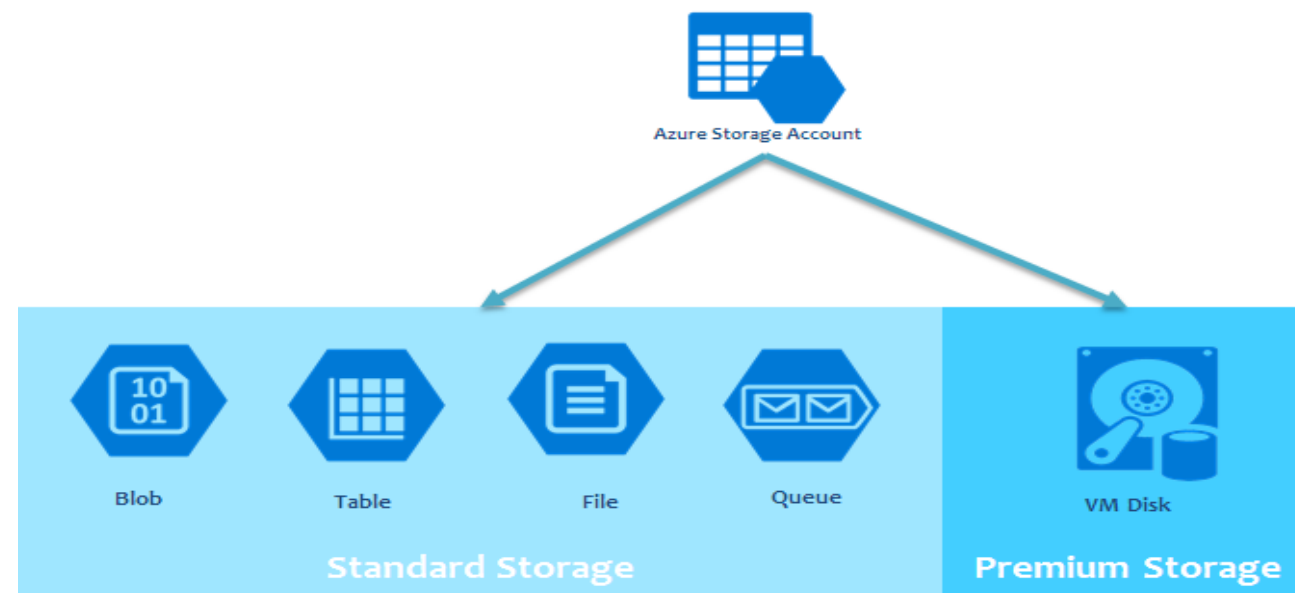


Storage Account

Azure Storage

It is a cloud storage platform designed for modern data storage scenarios.

A massively scalable object store for data objects, as well as disk storage for Azure virtual machines, is available via core storage services (VMs).



The service encrypts all data written to an Azure storage account.

Azure Storage Services

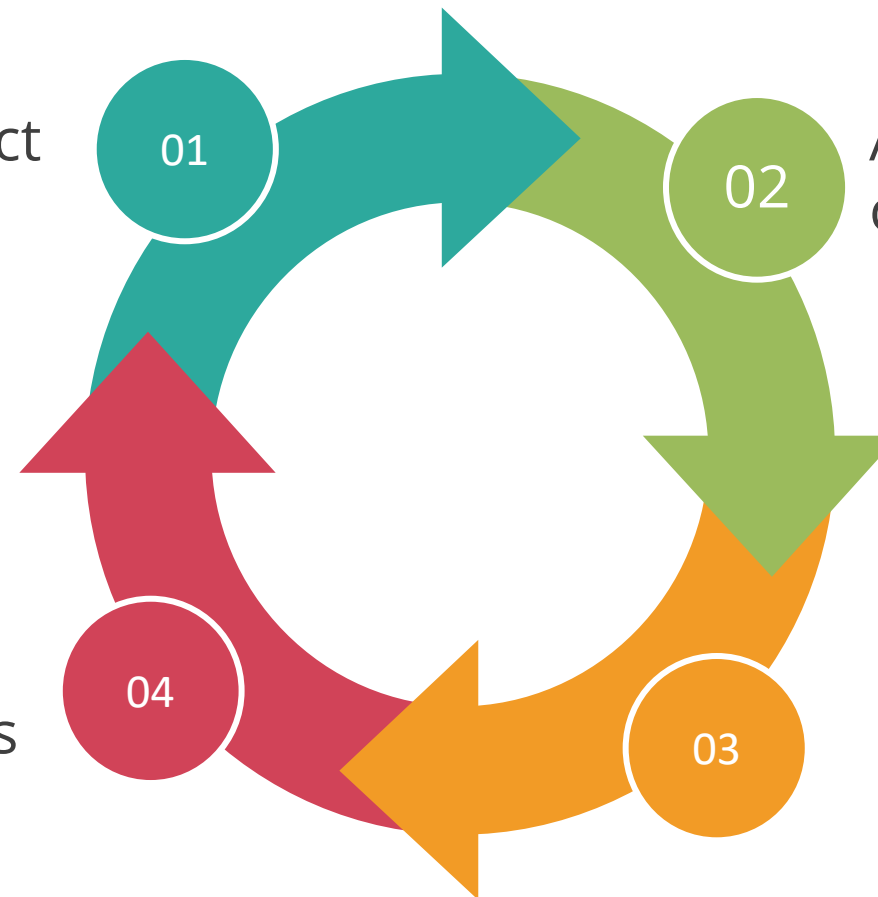
These are the Azure Storage services:

Azure Blobs
A text and binary data object store with huge scalability

Azure Files
A file share is available for cloud or on-premises deployments

Azure Queues
A messaging store that allows application components to communicate reliably

Azure Tables
A schemaless NoSQL store for structured data storage

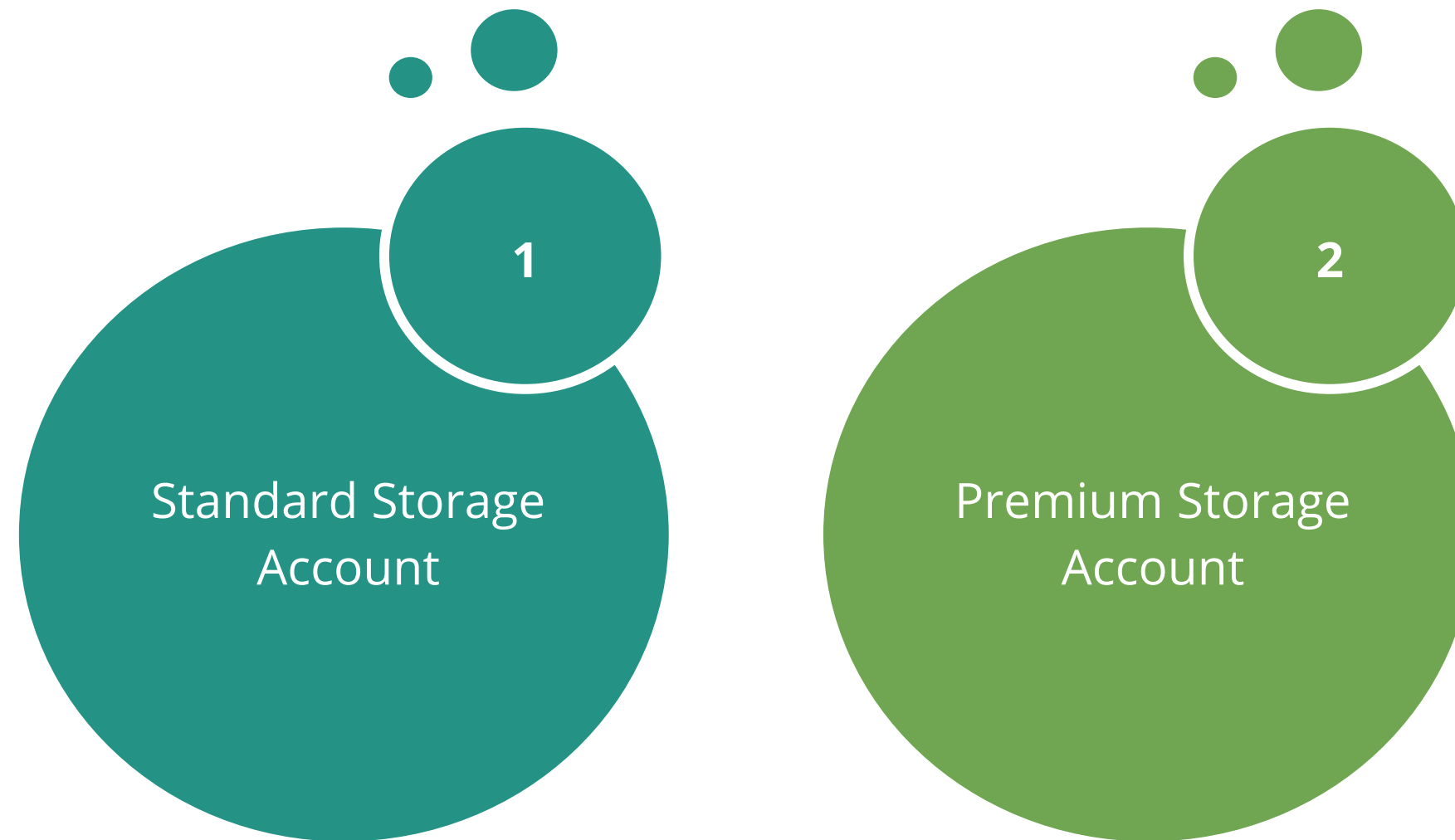


Azure Storage Services

Storage Services	Description
Azure Blobs	Allow unstructured data to be stored and accessed as block blobs on a huge scale
Azure Files	Allow accessing fully managed cloud file shares from anywhere by using the industry standard Server Message Block (SMB) protocol
Azure Tables	Allow the user to store structured NoSQL data in the cloud and provides a schemaless key/attribute store
Azure Queues	Allow asynchronous message queuing between application components

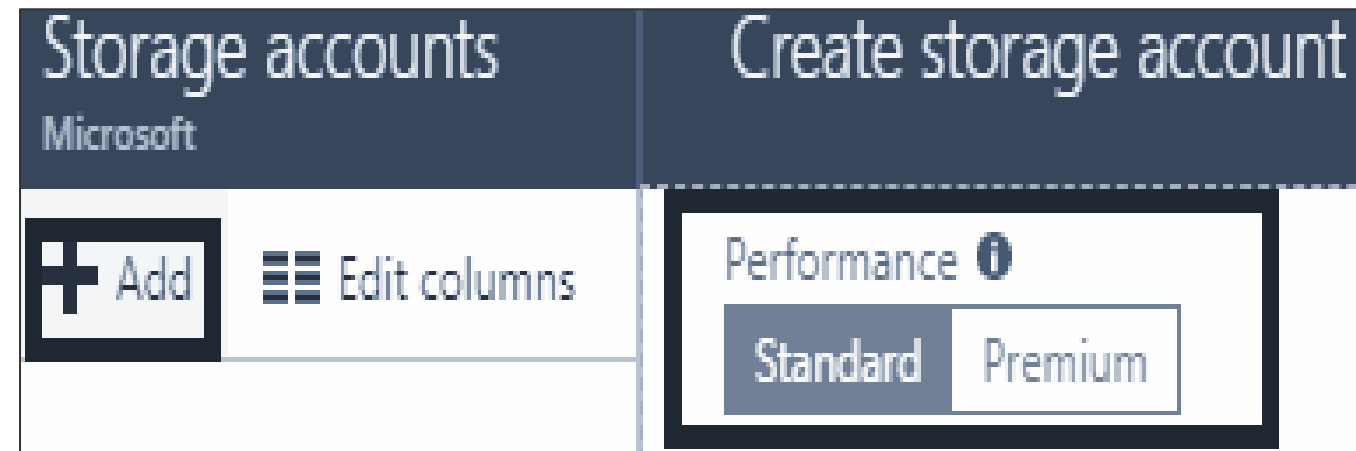
Standard and Premium Storage Accounts

Storage Accounts are of two types:



Standard Storage Account

The standard storage performance tier can store Tables, Queues, Files, Blobs, and Azure virtual machine disks.



Most apps use standard storage, which is less expensive and slower.

Premium Storage Account

Azure virtual machine disks are supported by the Premium storage performance tier only.

Create storage acc..

The cost of your storage account depends on the usage and the options [Learn more](#)

* Name ⓘ
anilpremiumaccount01 ✓
.core.windows.net

Deployment model ⓘ
Resource manager Classic

Account kind ⓘ
General purpose ▼

Performance ⓘ
Standard **Premium**

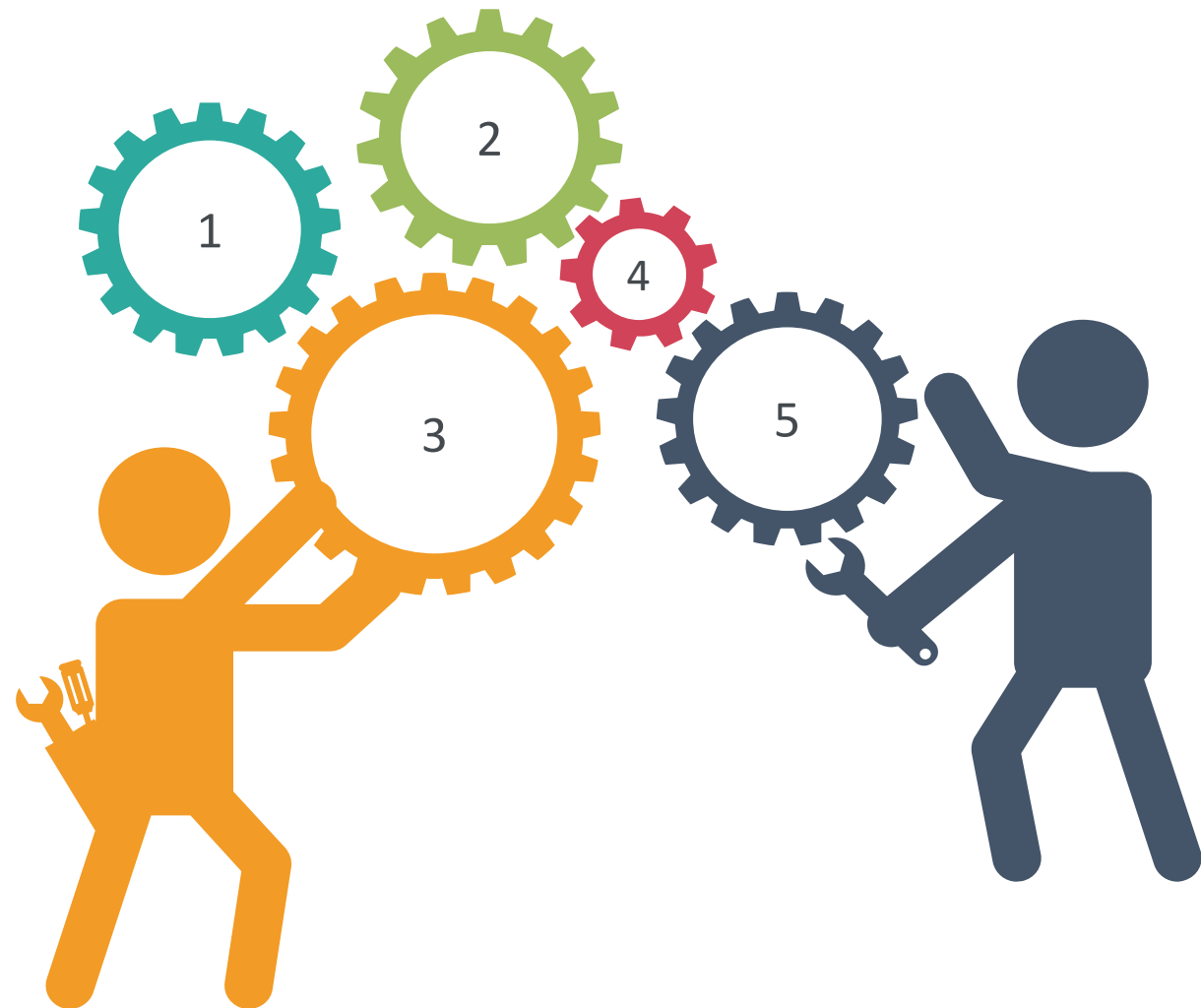
Replication ⓘ
Locally-redundant storage (LRS) ▼

* Subscription
Windows Azure MSDN - Visual Studio Ultir ▼

Transaction fees are not applied on premium disks.

Storage Types

There are five types of Storage:



Storage V2

Storage V1

Blob Storage

Block Blob Storage

File Storage

Storage Types

Storage types	Supported services	Supported performance tiers	Replication options
Blob storage	Blob (block blobs and append blobs only)	Standard	LRS, GRS, RA-GRS
General-purpose V1	Blob, File, Queue, Table, and Disk	Standard, Premium	LRS, GRS, RA-GRS
General-purpose V2	Blob, File, Queue, Table, and Disk	Standard, Premium	LRS, GRS, RA-GRS, ZRS, ZGRS (preview), RA-ZGRS (preview)
Block blob storage	Blob (block blobs and append blobs only)	Premium	LRS, ZRS (limited regions)
File storage	Files only	Premium	LRS, ZRS (limited regions)

Assisted Practice

Storage Account

Duration: 10 Min.

Problem Statement:

You've been asked to assist your organization with an Azure storage solution that can be used to store data objects such as blobs, file shares, queues, tables, and discs as an Azure Architect.

Assisted Practice: Guidelines

Steps to create a storage account:

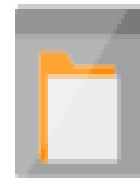
1. Logging in to the Azure portal at <https://portal.azure.com>
2. Selecting the subscription for which you want to create the storage account
3. Enabling Hierarchical namespace for Azure Data Lake Storage
4. Navigating to the storage account you want to access under the Settings section
5. Generating SAS and connection string



Storage Tiers

Blob Storage

It is an object storage solution for cloud, which is optimized for storing unstructured data.



Blobs

REST-based object storage for unstructured data

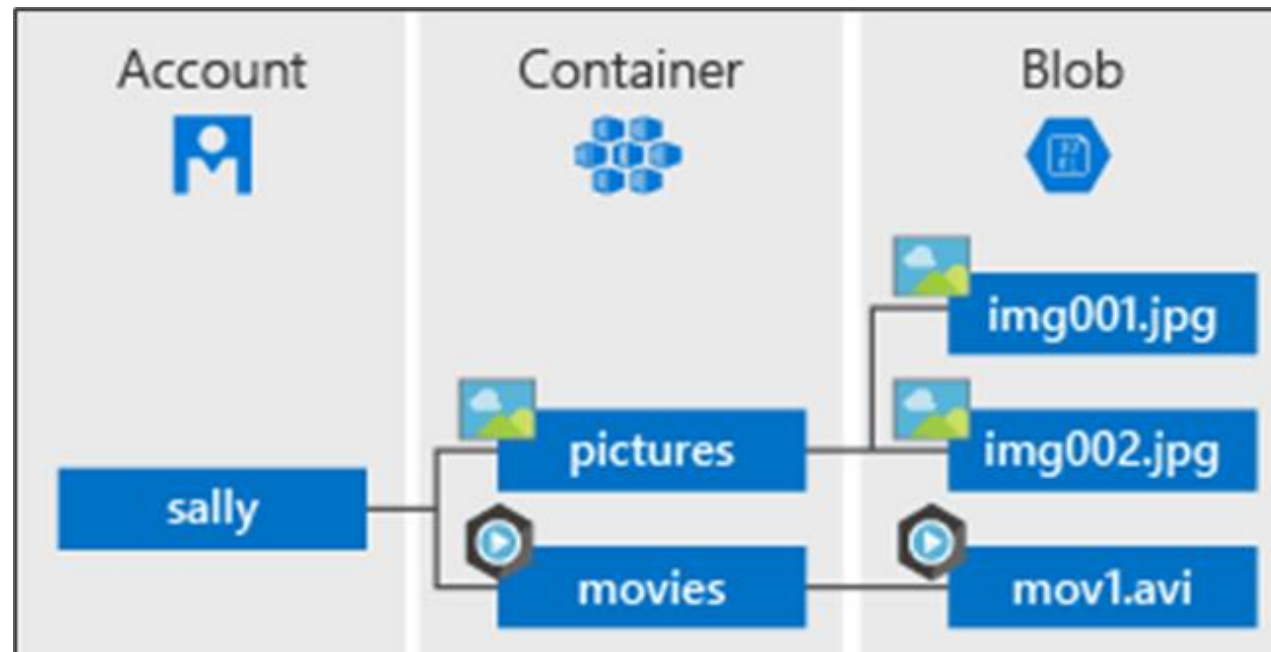
[Learn more](#)

[Explore data using Azure AD preview](#)

It can store any type of text or binary data.

Blob Storage

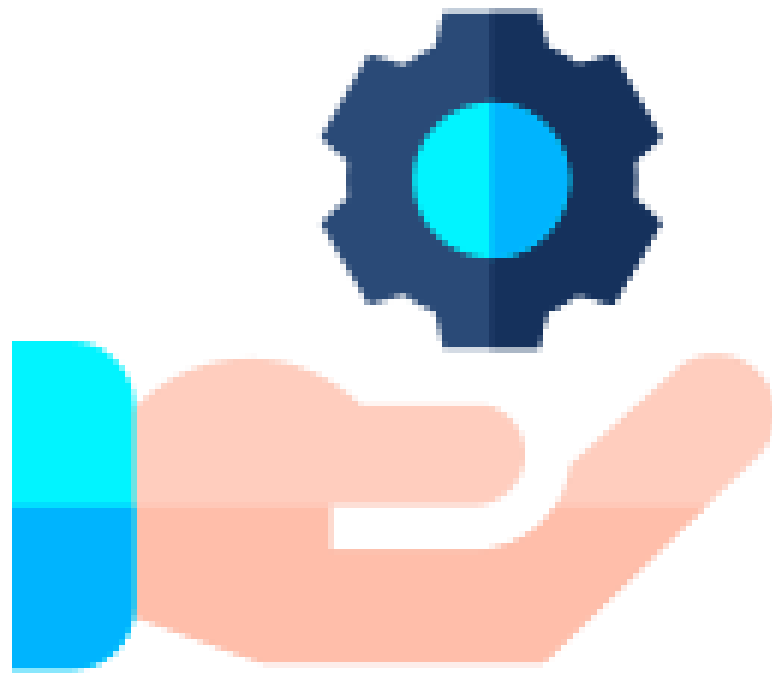
These are the common uses of blob storage:



- Serving images or documents directly to a browser
- Storing files for distributed access
- Streaming video and audio
- Storing data for backup and restore, disaster recovery, and archiving
- Storing data for analysis by an on-premises or Azure-hosted service

Blob Containers

These are the features of blob containers:



- A container organizes a set of blobs.
- A storage account can include an unlimited number of containers.
- A container can store an unlimited number of blobs.

Blob Performance Tiers

These are the blob performance tiers for storage:

Hot tier (inferred)

Optimized for frequent access of objects in the storage account

Cool tier

Optimized for storing large amounts of data that is infrequently accessed and stored for at least 30 days

Archive

Optimized for data that can tolerate several hours of retrieval latency and will remain in the Archive tier for at least 180 days

You can switch between these access tiers at any time.

Design for Blob Performance Tiers

The following points apply to the access tiers:



- At the account level, the hot and cool access tiers can be configured but the Archive access tier can't.
- During or after upload, the user can organize the hot, cool, and archive levels at the blob level.
- The data in the cool access tier can withstand a minor reduction in availability.
- The archive storage stores data offsite and offers the most cost-effective storage options.

Support Tiering for Storage Accounts

These are the usage scenarios for different tiers:

Hot tier (inferred)

Actively used data or data that is expected to be read from and written to regularly

Cool tier

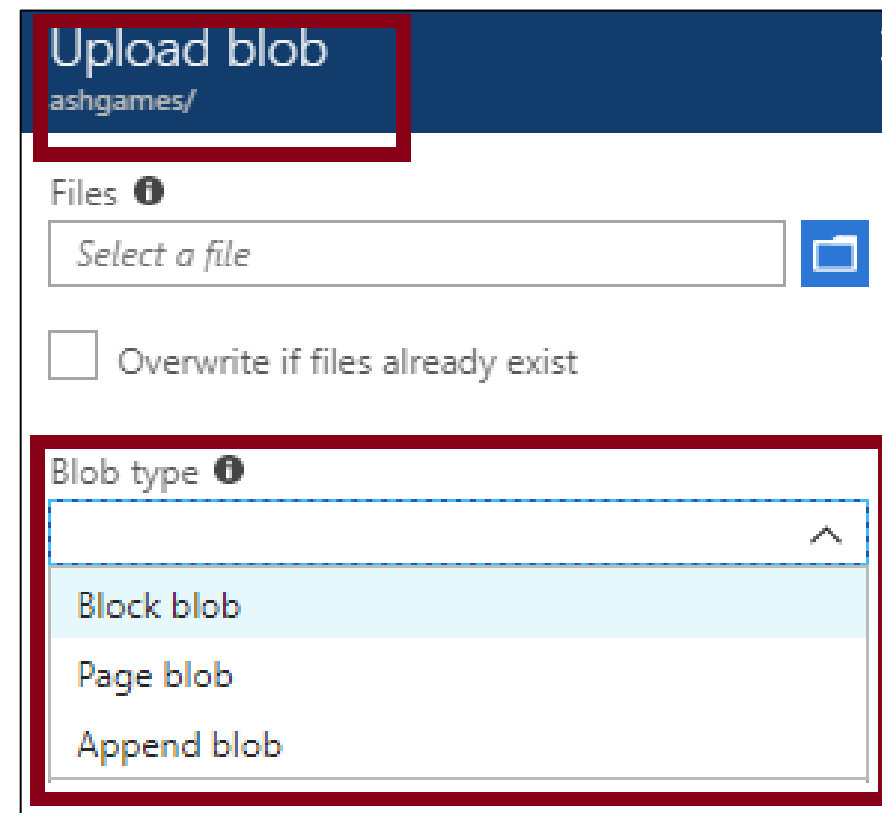
Backup and catastrophe recovery on a short-term basis

Archive

Datasets for long-term backup, secondary backup, and archiving

Uploading Blobs

These are the blob types that the user can upload:



The screenshot shows a web interface for uploading blobs. At the top, there's a dark blue header with the text 'Upload blob' and 'ashgames/' below it. Below the header, there's a section labeled 'Files' with an information icon. It contains a text input field with the placeholder 'Select a file' and a blue folder icon to its right. Below this is a checkbox labeled 'Overwrite if files already exist'. Further down, there's a section labeled 'Blob type' with an information icon. A dropdown menu is open, showing three options: 'Block blob', 'Page blob', and 'Append blob'. The 'Block blob' option is highlighted with a light blue background. The entire interface is framed by a dark red border.

- **Block blobs (default):** Useful for storing text or binary files
- **Page blobs:** More efficient for frequent read/write operations
- **Append blobs:** Useful for logging scenarios

User cannot change a blob type once it has been created

Assisted Practice

**Blob Storage
Min.**

Duration: 10

Problem Statement:

You've been asked to assist your organization with an Azure storage solution for storing vast volumes of unstructured data as an Azure Architect.

Assisted Practice: Guidelines

Steps to create blob storage:

1. Log into the Azure portal at <https://portal.azure.com>
2. Navigate to your new storage account on the Azure portal
3. Create a blob storage
4. Upload a block blob



Storage Access Solutions

Accessing Storage



- A storage account gives the user's data its own IP address.
- Every item the user saves in Azure Storage has a unique account name as part of its address.
- The endpoints for a user's storage account are made up of the account name and the Azure Storage service endpoint.
- The user can also configure a custom domain name:
 - CNAME: blobs.contoso.com
 - Command: Blobs.icloud.contoso.

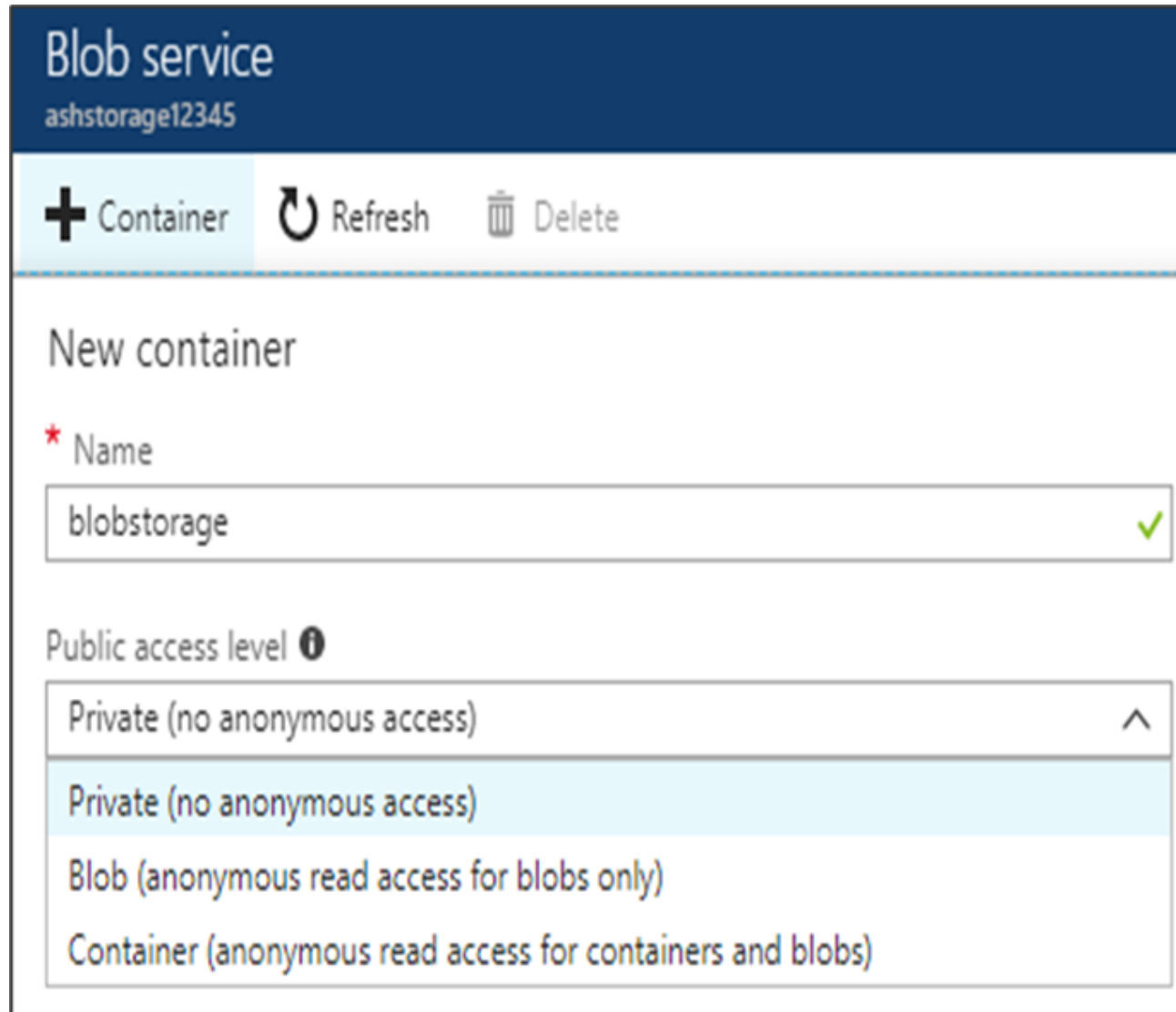
Accessing Storage

The endpoint format for each Azure Storage service is listed in the table below:

Storage Services	Endpoint
Azure Blobs	https://<storage-account>.blob.core.windows.net
Azure Files	https://<storage-account>.file.core.windows.net
Azure Tables	https://<storage-account>.table.core.windows.net
Azure Queues	https://<storage-account>.queue.core.windows.net

Blob Access Levels

These are the few access levels for Blob storage account:



The screenshot shows the 'New container' form in the Azure Blob service. The form has a header 'Blob service' and 'ashstorage12345'. Below the header are buttons for '+ Container', 'Refresh', and 'Delete'. The 'Name' field is filled with 'blobstorage' and has a green checkmark. The 'Public access level' dropdown menu is open, showing four options: 'Private (no anonymous access)' (selected), 'Private (no anonymous access)', 'Blob (anonymous read access for blobs only)', and 'Container (anonymous read access for containers and blobs)'.

Public access level
Private (no anonymous access)
Private (no anonymous access)
Blob (anonymous read access for blobs only)
Container (anonymous read access for containers and blobs)

Private blobs

No anonymous access

Blob access

Anonymous public read access for blobs only

Container access

Anonymous public read and list access to the entire container, including the blobs

Blob Access Policies

These are few policies to access the Blob storage account:

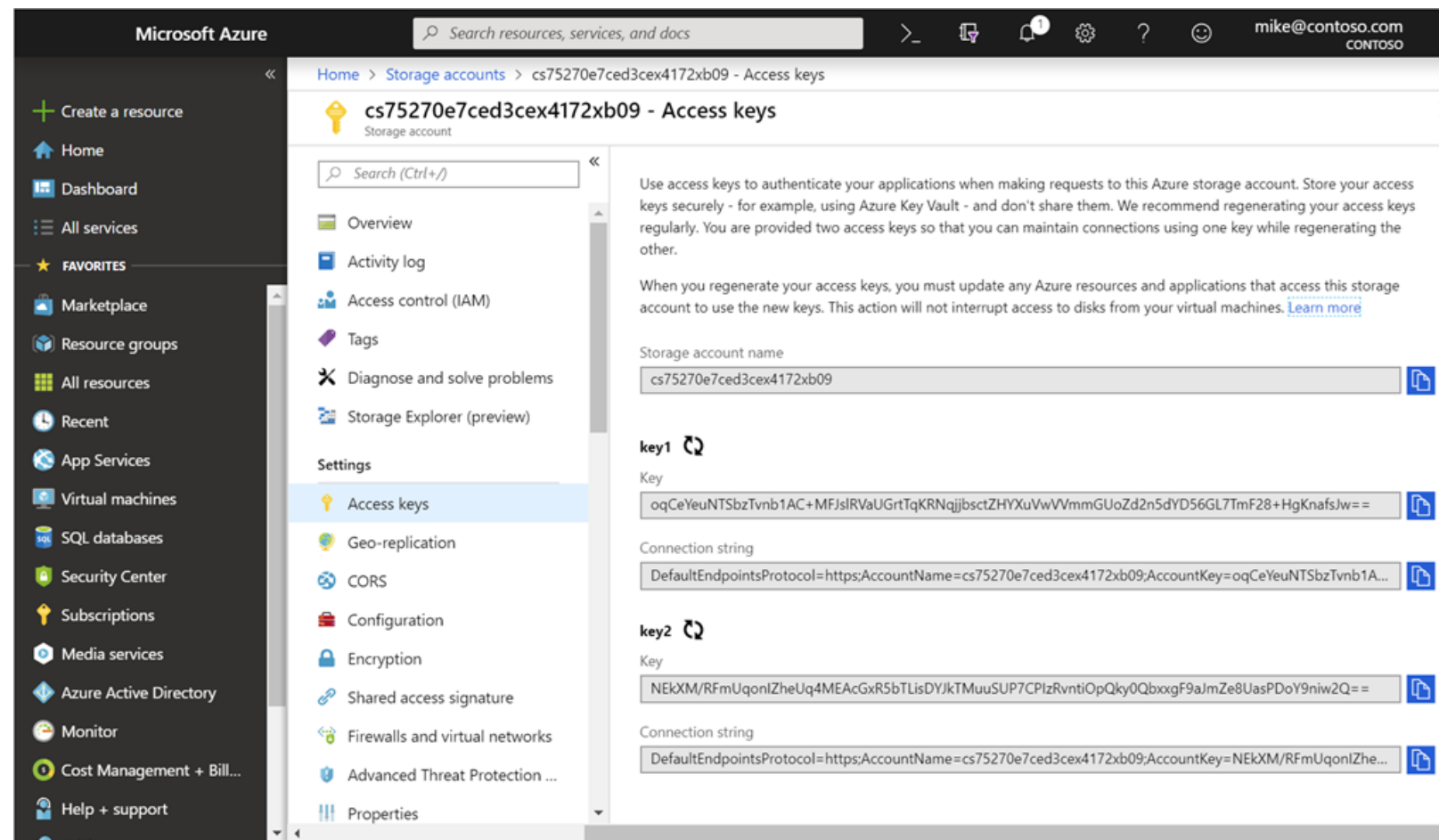
The screenshot shows the 'scripts - Access policy' page in the Azure portal. The left sidebar includes 'Overview', 'Access Control (IAM)', and 'Settings'. Under 'Settings', 'Access policy' is highlighted. The main content area is titled 'Add policy' and features a 'Save' button at the top left. Below it is an 'Identifier' input field. To the right of the identifier is a 'Permissions' section with a list of checkboxes: Read, Add, Create, Write, Delete, and List. At the bottom, there are 'Start time' and 'Expiry time' fields, each with a date picker (YYYY-MM-DD) and a time zone dropdown (h:mm:ss A, UTC-07:00 --- Current Time Zone...).

Policies

- Provides an additional level of control over server-side SAS
- Groups SAS to provide additional restrictions for signatures bound to the policy
- Supported for Blob containers, File shares, Tables, and Queues

Storage Account Keys

- Azure creates two keys, primary and secondary for each storage account
- Either of the keys provides full access to the account
- Keys should be regenerated on a regular basis or if they are compromised



Shared Access Signature (SAS)

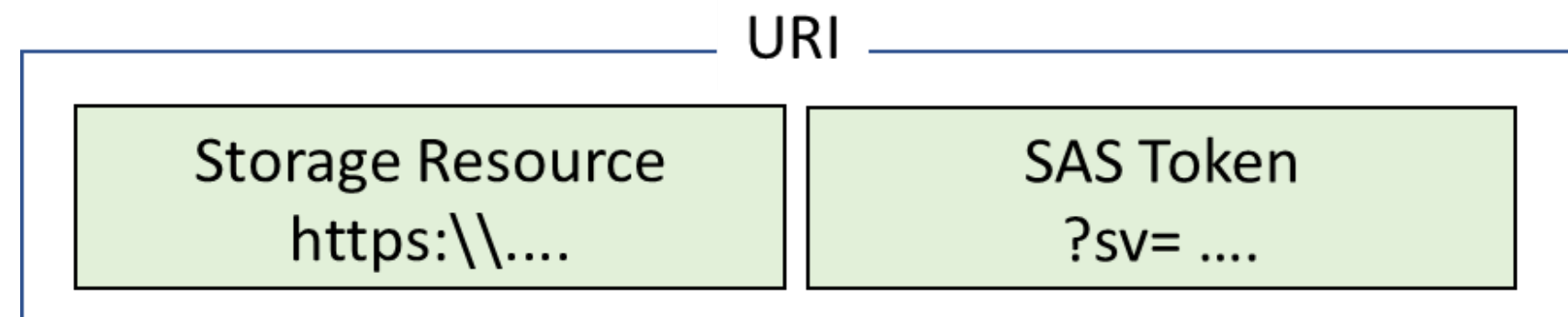
SAS provides delegated access to resources and also grants access to clients without sharing storage account keys.



The account SAS delegates access to resources in one of the storage services that are Blob, Queue, Table, or File service.

URI and SAS Parameters

- A SAS is a signed URI that points to one or more storage resources
- It consists of a storage resource URI and the SAS token



```
https://myaccount.blob.core.windows.net/?restype=service&comp=properties&sv=2015-04-05&ss=bf&srt=s&st=2015-04-29T22%3A18%3A26Z&se=2015-04-30T02%3A23%3A26Z&sr=b&sp=rw&sip=168.1.5.60-168.1.5.70&spr=https&sig=F%6GRVAZ5Cdj2Pw4txxxxx
```

Includes parameters for resource URI, storage services version, services, resource types, start time, expiry time, resource, permissions, IP range, protocol, and signature

Azure AD for Blobs and Queues

Overview

- 1 When a security principal attempts to access a blob or queue resource, the request must be authorized
- 2 The authentication step requires that an application request an OAuth 2.0 access token during runtime
- 3 The authorization step requires that one or more RBAC roles be assigned to the security principal

RBAC Roles for Access

These are the Built-in RBAC roles for access:

Blobs

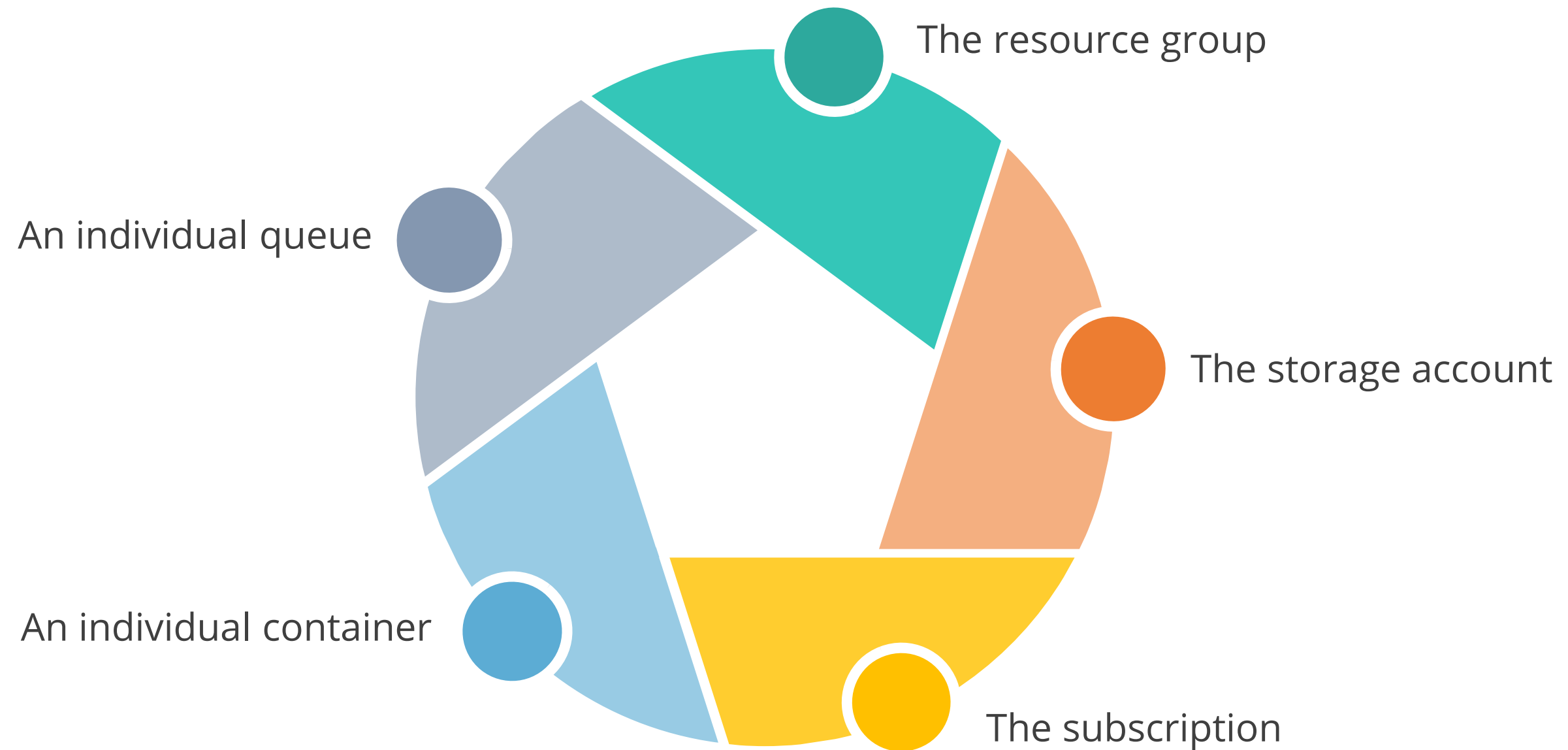
- Storage Blob Data Owner
- Storage Blob Data Contributor
- Storage Blob Data Reader

Queues

- Storage Queue Data Contributor
- Storage Queue Data Reader
- Storage Queue Data Message Processor
- Storage Queue Data Message Sender

Resource Scope

The levels at which the user can scope access to Azure blob and queue resources:



Access Data with an Azure AD Account

Access to blob or queue data from the Azure portal is available based on:

1

One of the storage account access keys (requires an RBAC role with Microsoft Storage/storage accounts/list keys/action permission)



2

An Azure AD account with permissions to access blob or queue data and to navigate through storage account resources

Assisted Practice

Generate SAS for Storage Account

Duration: 10 Min.

Problem Statement:

You've been asked to assist your organization with an Azure storage solution that can be used to offer limited access to containers and blobs in your storage account as an Azure Architect.

Assisted Practice: Guidelines

Steps to generate SAS for a storage account:

1. Log in to the Azure portal at <https://portal.azure.com>
2. Navigate to the storage account you want to access under the Settings section
3. Click on the shared access signature
4. Select the allowed resource types
5. Generate SAS for the storage account



Storage Management Tools

Manage Tiered Storage Using Azure Tools

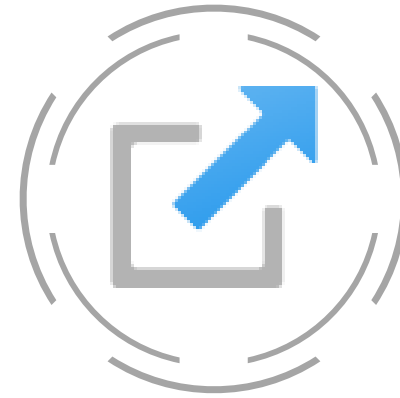
There are various tools available to manage Azure Storage:



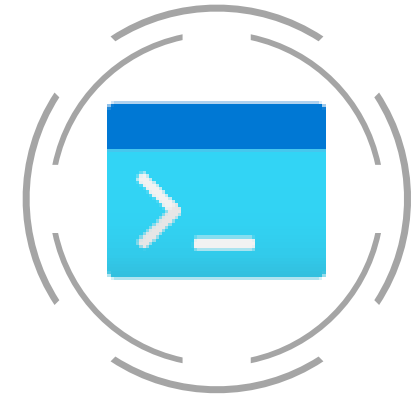
Azure Portal



Azure Storage Explorer



Azure CLI



Azure PowerShell

Manage Tiered Storage Using Azure Tools

The following commands are used to manage the Azure tools:

```
az storage blob upload
az storage blob list
az storage blob download
az storage blob set-tier
```

```
$Storage= "StorageAccountName"
$Key = "Storage Account Key"
$Container = "Blob Container"
$Blobs = Get-AzureStorageblob -Container $Container
$blob.icloudblob.setstandardblobtier("Cool")
```

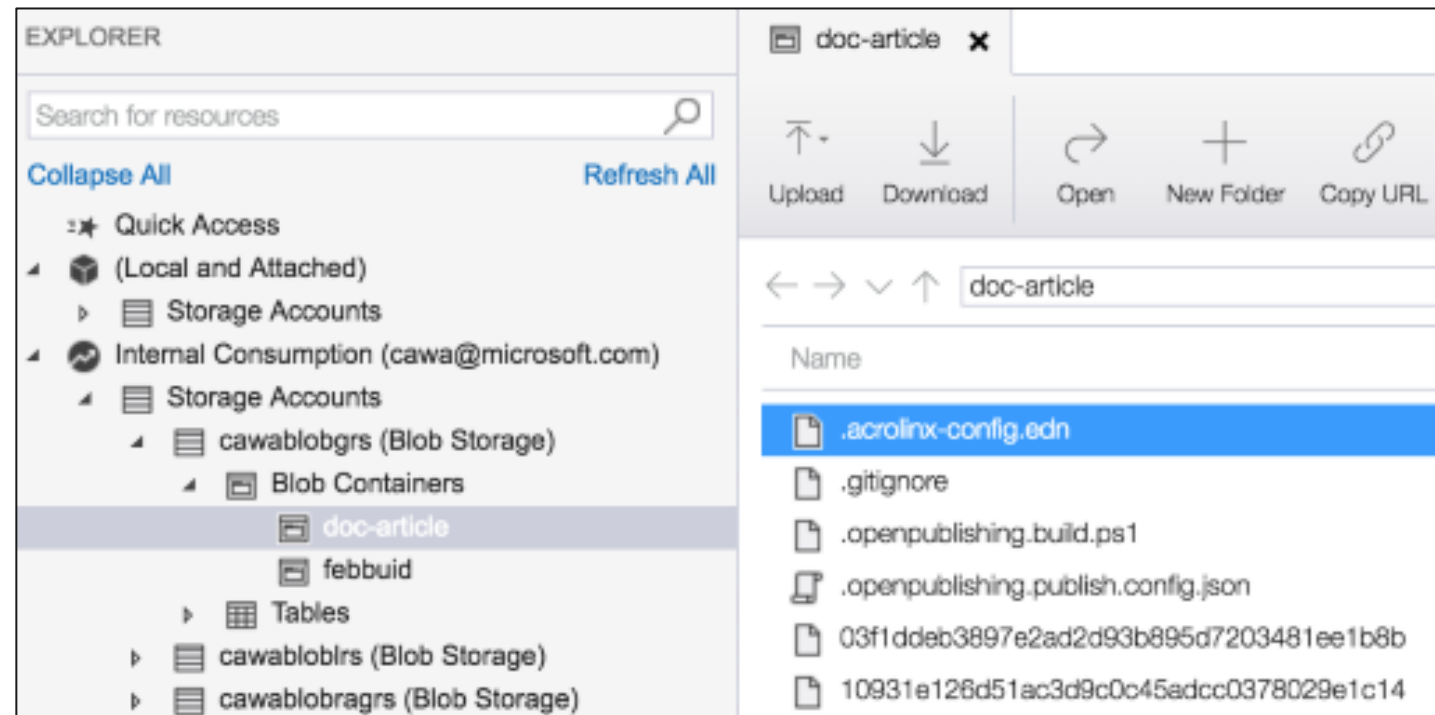
Comparing Tools for Working with Azure Storage

The table given below shows the comparison of tools that can run on various platforms:

Azure Storage Client Tool	Supported Platform	Block Blob	Page Blob	Append Blob	Tables	Queues	Files
Azure Portal	Web	Yes	Yes	Yes	Yes	Yes	Yes
Azure Storage Explorer	Windows, OSX	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Visual Studio Cloud Explorer	Windows	Yes	Yes	Yes	Yes	Yes	No

Azure Storage Explorer

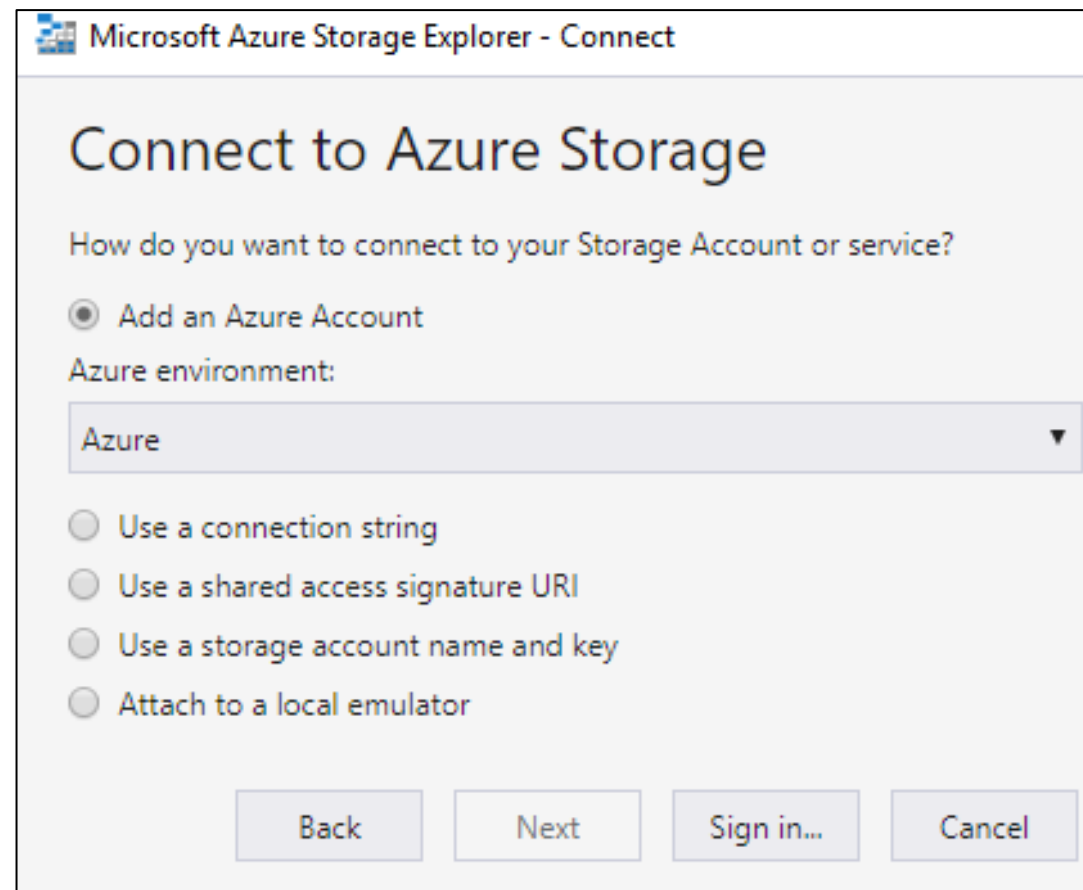
Azure Storage Explorer is a single software for Windows, macOS, and Linux that makes working with Azure Storage data simple.



- Access multiple accounts and subscriptions
- Create, delete, view, and edit storage resources
- View and edit Blob, Queue, Table, File, Cosmos DB storage, and Data Lake Storage
- Obtain shared access signature (SAS) keys
- Available for Windows, Mac, and Linux

Azure Storage Connection

The procedure for connecting to Azure storage is given below:



The screenshot shows the 'Microsoft Azure Storage Explorer - Connect' dialog box. The title bar reads 'Microsoft Azure Storage Explorer - Connect'. The main heading is 'Connect to Azure Storage'. Below this, it asks 'How do you want to connect to your Storage Account or service?'. There are five radio button options: 'Add an Azure Account' (which is selected), 'Use a connection string', 'Use a shared access signature URI', 'Use a storage account name and key', and 'Attach to a local emulator'. Below the radio buttons, there is a dropdown menu labeled 'Azure environment:' with 'Azure' selected. At the bottom, there are four buttons: 'Back', 'Next', 'Sign in...', and 'Cancel'.

Options

- Connect to an Azure subscription
- Work with local development storage (emulator)
- Attach to external storage
- Attach a storage account or storage service by using a shared access signature
- Connect to an Azure Cosmos DB account by using a connection string

Assisted Practice

Manage Storage Access Keys Duration: 10 Min.

Problem Statement:

Demonstrate an Azure storage solution that can be used to employ Shared Key authorization to authorize access to data in your storage account.

Assisted Practice: Guidelines

Steps to manage storage access keys:

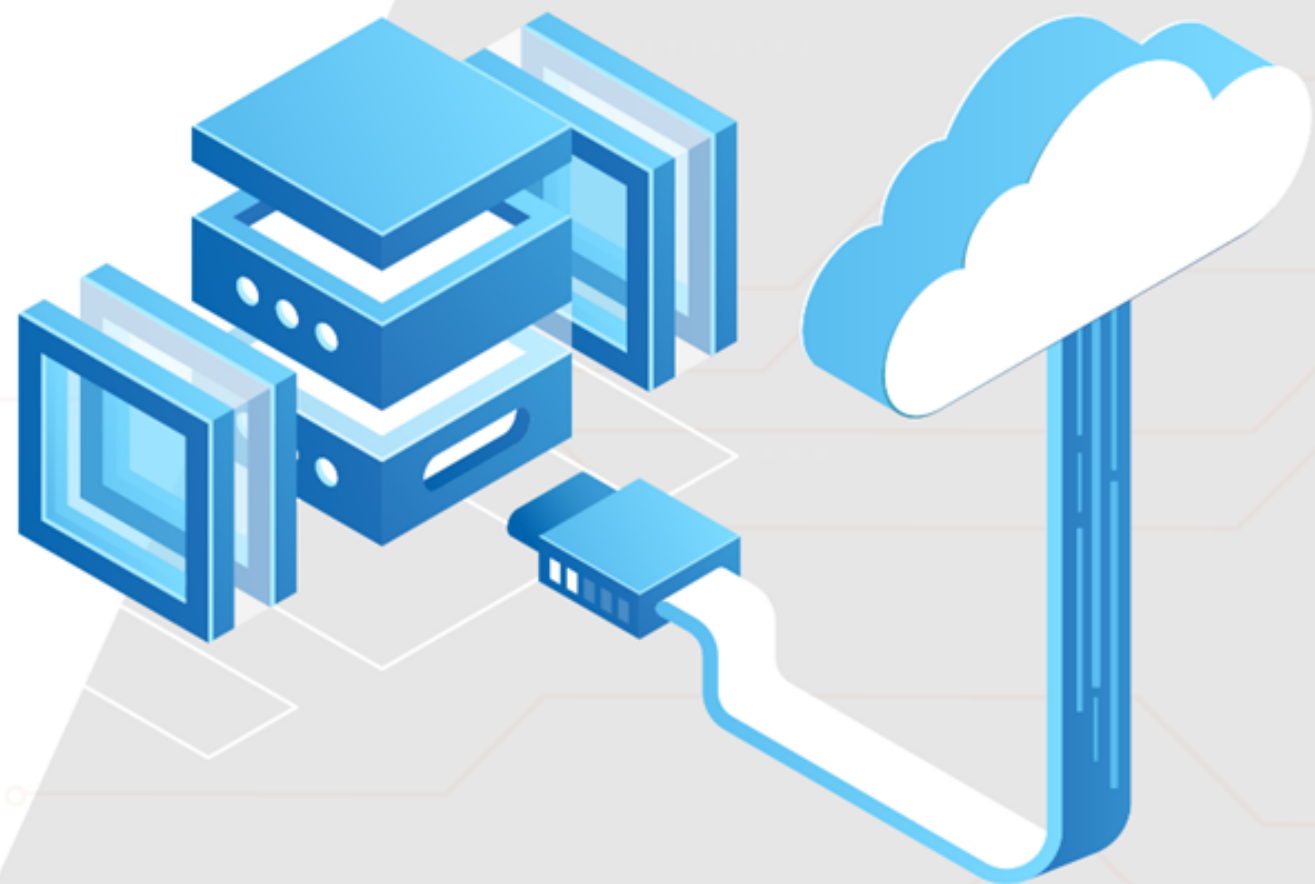
1. Login to the Azure portal
2. Search for and select Storage Account
3. Manage storage access keys



Key Takeaways

- Azure storage is a cloud storage platform designed for modern data storage scenarios, it provides services like Azure Blobs, Files, Queues, and Tables.
- Blob Storage is an object storage solution for cloud, which is optimized for storing unstructured data.
- SAS provides access to resources and also grants access to clients without sharing storage account keys.
- Azure Storage Explorer is a single software for all operating systems that makes working with Azure Storage data simple.





Thank you