

# Introduction to Azure Storage

- The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud.
- Azure Storage data objects are accessible from anywhere in the world over HTTP or HTTPS via a REST API. Azure Storage also offers client libraries for developers building applications or services with .NET, Java, Python, JavaScript, C++, and Go.
- Developers and IT professionals can use Azure PowerShell and Azure CLI to write scripts for data management or configuration tasks. The Azure portal and Azure Storage Explorer provide user-interface tools for interacting with Azure Storage.
- Link : <https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

# Benefits of Azure Storage

- Azure Storage services offer the following benefits for application developers and IT professionals:
- **Durable and Highly available:** Built-in redundancy keeps data safe in the event of underlying hardware failures within the Azure infrastructure.

In addition to this redundancy, you can also cross multiple data centers and even multiple geographical regions. This provides protection against natural disasters or local failures at the data center level.

You can also opt to replicate data across data centers or geographical regions for additional protection from local catastrophe or natural disaster. Data replicated in this way remains highly available in the event of an unexpected outage.

- **Secure/Encrypted:** All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- **Scalable:** Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Managed:** Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible:** Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides client libraries for Azure Storage in a variety of languages, including .NET, Java, Node.js, Python, PHP, Ruby, Go, and others, as well as a mature REST API.

Azure Storage supports scripting in **Azure PowerShell or Azure CLI**. And the **Azure portal and Azure Storage Explorer** offer easy visual solutions for working with your data.

# Azure Storage data services

- **The Azure Storage platform includes the following data services:**

**1) Azure Blobs:** Azure blob storage provides object storage for the cloud. This offering has been optimized to support massive amounts of unstructured data. Unstructured data is data that does not fit a specific data model. For example, text and binary data would be the type of unstructured data that falls under blob storage. It also includes support for big data analytics through Data Lake Storage Gen2.

- Link : <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>

**2) Azure Files:** Azure Files is essentially a fully managed file share system available in the cloud. You can access Azure Files through the typical SMB protocol. ([Server Message Block \(SMB\) protocol](#) or [Network File System \(NFS\) protocol](#)) . You can mount Azure file shares from Windows, Linux, and MacOS machines that reside both on-prem and in the cloud. You can even cash Azure file shares on Windows servers, using the Azure file sync service. This helps make data more accessible for remote offices. Managed file shares for cloud or on-premises deployments.

- Link : <https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction>

**3) Azure Queues:** Azure queue storage is a special type of storage service. It's not meant for storing files. Instead, it's designed for storing large numbers of **messages**. These messages are used in communications between the different components of a distributed application. Such messages can be accessed from anywhere in the world through authenticated calls via HTTP or HTTPS.

- Each queue message can be up to 64 KB in size. A typical queue can contain millions of messages. A messaging store for reliable messaging between application components.
- Link : <https://docs.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction>

**4) Azure Tables:** Azure table storage is a storage offering intended for the storage of structured NoSQL data. It offers a key/attribute store and a schema-less design. The schema-less design of Azure table storage allows you to more easily adapt the data to the needs of your business or application. A NoSQL store for schemaless storage of structured data.

- Link : <https://docs.microsoft.com/en-us/azure/storage/tables/table-storage-overview>

**5) Azure Disks:** Azure managed disks are essentially block-level storage volumes. These storage volumes, which are managed by Azure, are used, to provide storage capabilities for **virtual machines**. A managed disk is much like a physical disk that you would see in an on-prem server. However, it's virtualized. There are several types of disks available in Azure. They include **ultra disks, premium SSD disks, standard SSD disks and standard HDD disks**. We will touch on these in more detail later on.

- **Block-level storage volumes for Azure VMs.**
- Link : <https://docs.microsoft.com/en-us/azure/virtual-machines/managed-disks-overview>

Feature	Description	When to use
<b>Azure Files</b>	<p>Offers fully managed cloud file shares that you can access from anywhere via the industry standard Server Message Block (SMB) protocol.</p> <p>You can mount Azure file shares from cloud or on-premises deployments of Windows, Linux, and macOS.</p>	<p>You want to "lift and shift" an application to the cloud that already uses the native file system APIs to share data between it and other applications running in Azure.</p> <p>You want to replace or supplement on-premises <b>file servers</b> or <b>NAS devices</b>.</p> <p>You want to store development and debugging tools that need to be accessed from many virtual machines.</p>
<b>Azure Blobs</b>	<p>Allows unstructured data to be stored and accessed at a massive scale in block blobs.</p> <p>Also supports <a href="#">Azure Data Lake Storage Gen2</a> for enterprise big data analytics solutions.</p>	<p>You want your application to <b>support streaming and random-access scenarios</b>.</p> <p>You want to be able to access application data from anywhere.</p> <p>You want to build an enterprise data lake on Azure and perform big data analytics.</p>
<b>Azure Disks</b>	<p>Allows data to be persistently stored and accessed from an attached virtual hard disk.</p>	<p>You want to "lift and shift" applications that use native file system APIs to read and write data to persistent disks.</p> <p><b>You want to store data that is not required to be accessed from outside the virtual machine to which the disk is attached.</b></p>
<b>Azure Queues</b>	<p>Allows for asynchronous message queueing between application components.</p>	<p>You want to decouple application components and use asynchronous messaging to communicate between them.</p> <p>For guidance around when to use Queue storage versus Service Bus queues, see <a href="#">Storage queues and Service Bus queues - compared and contrasted</a>.</p>
<b>Azure Tables</b>	<p>Allow you to store structured NoSQL data</p>	<p>You want to store flexible datasets like user data <b>for web applications</b></p>



D



E



# Storage accounts

- A storage account provides a unique namespace in Azure for your data. Every object that you store in Azure Storage has an address that includes your **unique storage account name**. The combination of the account name and the Blob Storage endpoint forms the base address for the objects in your storage account.
- For example, if your storage account is named ***mystorageaccount***, then the default endpoint for Blob storage is:
- **<http://mystorageaccount.blob.core.windows.net>**

## Storage Account Types in Azure

STORAGE ACCOUNT TYPE	SUPPORTED SERVICES	SUPPORTED PERFORMANCE TIERS	SUPPORTED ACCESS TIERS	REPLICATION OPTIONS	ENCRYPTION
<b>GENERAL-PURPOSE V2</b>	Blob, File, Queue, Table, Disk, and Data Lake Gen2	Standard Premium	Hot Cool Archive	LRS, GRS, RA-GRS, ZRS, GZRS (preview), RA-GZRS (preview)	Encrypted
<b>GENERAL-PURPOSE V1</b>	Blob, File, Queue, Table, and Disk	Standard Premium	N/A	LRS, GRS, RA-GRS	Encrypted
<b>BLOCK BLOB STORAGE</b>	Blob (block blobs and append blobs only)	Premium	N/A	LRS, ZRS	Encrypted
<b>FILE STORAGE</b>	File only	Premium	N/A	LRS, ZRS	Encrypted
<b>BLOB STORAGE</b>	Blob (block blobs and append blobs only)	Standard	Hot Cool Archive	LRS, GRS, RA-GRS	Encrypted

## DEMO : Create a storage account

- A storage account is an Azure Resource Manager resource. Resource Manager is the deployment and management service for Azure
- Every Resource Manager resource, including an Azure storage account, must belong to an Azure resource group. A resource group is a logical container for grouping your Azure services. When you create a storage account, you have the option to either create a new resource group, or use an existing resource group. This how-to shows how to create a new resource group.
- Link : <https://docs.microsoft.com/en-us/azure/storage/common/storage-account-create?toc=%2Fazure%2Fstorage%2Fblobs%2Ftoc.json&tabs=azure-portal>
- Use above link to explain all details.
- Create Azure Storage account and its key configuration items.



# Introduction to Azure Blob storage

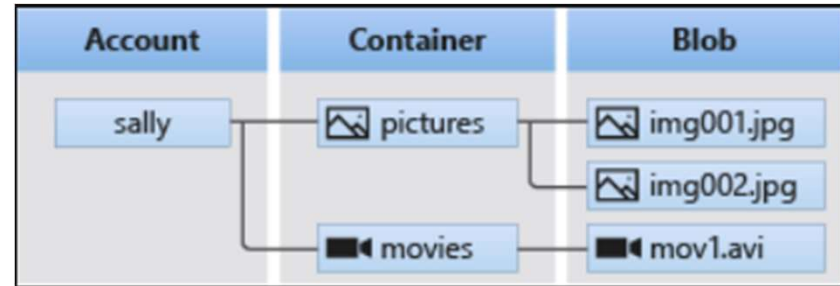
- Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.
- **Blob storage is designed for :**
- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Writing to log files.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.
- Link : <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>

➤ **About Azure Data Lake Storage Gen2**

- Blob storage supports Azure Data Lake Storage Gen2, Microsoft's enterprise big data analytics solution for the cloud. Azure Data Lake Storage Gen2 offers a hierarchical file system as well as the advantages of Blob storage, including:
- Low-cost, tiered storage
- High availability
- Strong consistency
- Disaster recovery capabilities

# Blob storage resources

- Blob storage offers three types of resources:
- The **storage account**
- A **container** in the storage account
- A **blob** in a container.
- The following diagram shows the relationship between these resources
- The storage account creates the unique namespace in Azure that you use to access your data. When you access your data, you'll use a combination of the storage account name and the Azure Storage blob endpoint that you are trying to access. Together, these two pieces form the base address for the objects that reside in the storage account.



The following table describes the different types of storage accounts that are supported for **Blob Storage**:

Type of storage account	Performance tier	Usage
General-purpose v2	Standard	Standard storage account type for blobs, file shares, queues, and tables. Recommended for most scenarios using Blob Storage or one of the other Azure Storage services.
Block blob	Premium	Premium storage account type for block blobs and append blobs. Recommended for scenarios with high transaction rates or that use smaller objects or require consistently low storage latency. <a href="#">Learn more about workloads for premium block blob accounts...</a>
Page blob	Premium	Premium storage account type for page blobs only. <a href="#">Learn more about workloads for premium page blob accounts...</a>

# Containers

- **A container organizes a set of blobs, similar to a directory in a file system. A storage account can include an unlimited number of containers, and a container can store an unlimited number of blobs.**

- 

**Blobs** : Azure Storage supports three types of blobs:

- **Block blobs** store text and binary data. Block blobs are made up of blocks of data that can be managed individually. Block blobs can store up to about 190.7 TiB.
- **Append blobs** are made up of blocks like block blobs, but are optimized for append operations. Append blobs are ideal for scenarios such as logging data from virtual machines.
- **Page blobs** store random access files up to 8 TiB in size. Page blobs store virtual hard drive (VHD) files and serve as disks for Azure virtual machines.

# Move data to Blob storage

- A number of solutions exist for migrating existing data to Blob storage:
- **AzCopy** is an easy-to-use command-line tool for Windows and Linux that copies data to and from Blob storage, across containers, or across storage accounts. For more information about AzCopy, see [Transfer data with the AzCopy v10](#).
- The **Azure Storage Data Movement library** is a .NET library for moving data between Azure Storage services. The AzCopy utility is built with the Data Movement library. For more information, see the [reference documentation](#) for the Data Movement library.
- **Azure Data Factory** supports copying data to and from Blob storage by using the account key, a shared access signature, a service principal, or managed identities for Azure resources. For more information, see [Copy data to or from Azure Blob storage by using Azure Data Factory](#).
- **Blobfuse** is a virtual file system driver for Azure Blob storage. You can use blobfuse to access your existing block blob data in your Storage account through the Linux file system. For more information, see [How to mount Blob storage as a file system with blobfuse](#).
- **Azure Data Box** service is available to transfer on-premises data to Blob storage when large datasets or network constraints make uploading data over the wire unrealistic. Depending on your data size, you can request [Azure Data Box Disk](#), [Azure Data Box](#), or [Azure Data Box Heavy](#) devices from Microsoft. You can then copy your data to those devices and ship them back to Microsoft to be uploaded into Blob storage.
- The **Azure Import/Export service** provides a way to import or export large amounts of data to and from your storage account using hard drives that you provide. For more information, see [Use the Microsoft Azure Import/Export service to transfer data to Blob storage](#).

# DEMO : Create Azure Blob Storage

- Demo : Create Blob Container in Azure Storage Account and Upload photos.
- Link : <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-upload-process-images?tabs=dotnet%2Cazure-powershell>



# What is Azure Files?

- Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) protocol or Network File System (NFS) protocol.
- Azure Files file shares can be mounted concurrently by cloud or on-premises deployments. SMB Azure file shares are accessible from Windows, Linux, and macOS clients. NFS Azure Files shares are accessible from Linux or macOS clients. Additionally, SMB Azure file shares can be cached on Windows Servers with Azure File Sync for fast access near where the data is being used.
- Link : <https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction>
- Link : <https://docs.microsoft.com/en-us/windows/win32/fileio/microsoft-smb-protocol-and-cifs-protocol-overview>

# Why Azure Files is useful

- Azure file shares can be used to:
- **Replace or supplement on-premises file servers:**

Azure Files can be used to completely replace or supplement traditional on-premises file servers or NAS devices. Popular operating systems such as Windows, macOS, and Linux can directly mount Azure file shares wherever they are in the world. SMB Azure file shares can also be replicated with Azure File Sync to Windows Servers, either on-premises or in the cloud, for performance and distributed caching of the data where it's being used. With the recent release of [Azure Files AD Authentication](#), SMB Azure file shares can continue to work with AD hosted on-premises for access control.
- **"Lift and shift" applications:**

Azure Files makes it easy to "lift and shift" applications to the cloud that expect a file share to store file application or user data. Azure Files enables both the "classic" lift and shift scenario, where both the application and its data are moved to Azure, and the "hybrid" lift and shift scenario, where the application data is moved to Azure Files, and the application continues to run on-premises.
- **Simplify cloud development:**

Azure Files can also be used in numerous ways to simplify new cloud development projects. For example:

  -

- **Shared application settings:**

A common pattern for distributed applications is to have configuration files in a centralized location where they can be accessed from many application instances. Application instances can load their configuration through the File REST API, and humans can access them as needed by mounting the SMB share locally.

- **Diagnostic share:**

An Azure file share is a convenient place for cloud applications to write their logs, metrics, and crash dumps. Logs can be written by the application instances via the File REST API, and developers can access them by mounting the file share on their local machine. This enables great flexibility, as developers can embrace cloud development without having to abandon any existing tooling they know and love.

- **Dev/Test/Debug:**

When developers or administrators are working on VMs in the cloud, they often need a set of tools or utilities. Copying such utilities and tools to each VM can be a time consuming exercise. By mounting an Azure file share locally on the VMs, a developer and administrator can quickly access their tools and utilities, no copying required.

- **Containerization:**

Azure file shares can be used as persistent volumes for stateful containers. Containers deliver "build once, run anywhere" capabilities that enable developers to accelerate innovation. For the containers that access raw data at every start, a shared file system is required to allow these containers to access the file system no matter which instance they run on.

# Key benefits of Azure Files

- **Shared access.** Azure file shares support the industry standard SMB and NFS protocols, meaning you can seamlessly replace your on-premises file shares with Azure file shares without worrying about application compatibility. Being able to share a file system across multiple machines, applications/instances is a significant advantage with Azure Files for applications that need shareability.
- **Fully managed.** Azure file shares can be created without the need to manage hardware or an OS. This means you don't have to deal with patching the server OS with critical security upgrades or replacing faulty hard disks.
- **Scripting and tooling.** PowerShell cmdlets and Azure CLI can be used to create, mount, and manage Azure file shares as part of the administration of Azure applications. You can create and manage Azure file shares using Azure portal and Azure Storage Explorer.
- **Resiliency.** Azure Files has been built from the ground up to be always available. Replacing on-premises file shares with Azure Files means you no longer have to wake up to deal with local power outages or network issues.
- **Familiar programmability.** Applications running in Azure can access data in the share via file system I/O APIs. Developers can therefore leverage their existing code and skills to migrate existing applications. In addition to System IO APIs, you can use Azure Storage Client Libraries or the Azure Storage REST API.

# Create an Azure file share

To create an Azure file share, you need to answer three questions about how you will use it:



## 1. What are the performance requirements for your Azure file share?

Azure Files offers standard file shares which are hosted on hard disk-based (HDD-based) hardware, and premium file shares, which are hosted on solid-state disk-based (SSD-based) hardware.



## 2. What are your redundancy requirements for your Azure file share?

Standard file shares offer locally-redundant (LRS), zone redundant (ZRS), geo-redundant (GRS), or geo-zone-redundant (GZRS) storage, however the large file share feature is only supported on locally redundant and zone redundant file shares. Premium file shares do not support any form of geo-redundancy.



## 3. What size file share do you need?

In local and zone redundant storage accounts, Azure file shares can span up to 100 TiB, however in geo- and geo-zone redundant storage accounts, Azure file shares can span only up to 5 TiB.



Premium file shares are available with locally redundancy and zone redundancy in a subset of regions. To find out if premium file shares are currently available in your region, see the [products available by region](#) page for Azure. For information about regions that support ZRS, see [Azure Storage redundancy](#).

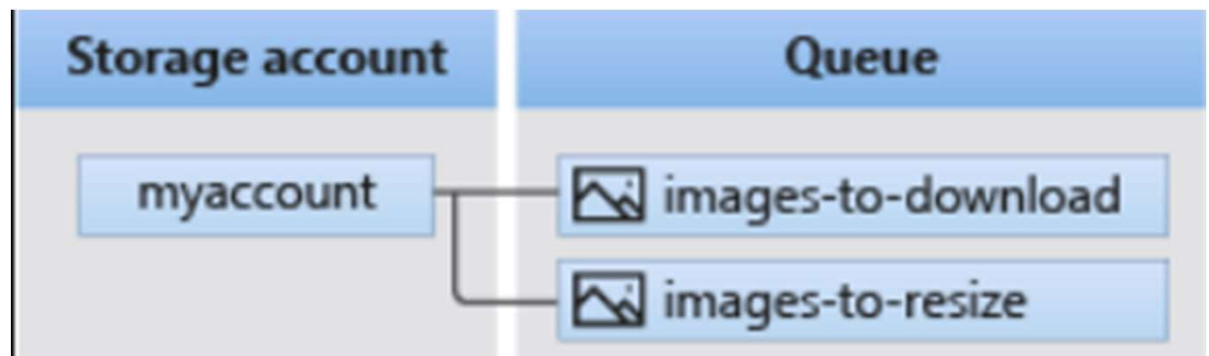
## **DEMO :** Create a storage account for Azure Files and upload files

- Azure supports multiple types of storage accounts for different storage scenarios customers may have, but there are two main types of storage accounts for Azure Files. Which storage account type you need to create depends on whether you want to create a standard file share or a premium file share:
- **General purpose version 2 (GPv2) storage accounts:** GPv2 storage accounts allow you to deploy Azure file shares on standard/hard disk-based (HDD-based) hardware. In addition to storing Azure file shares, GPv2 storage accounts can store other storage resources such as blob containers, queues, or tables. File shares can be deployed into the transaction optimized (default), hot, or cool tiers.
- **FileStorage storage accounts:** FileStorage storage accounts allow you to deploy Azure file shares on premium/solid-state disk-based (SSD-based) hardware. FileStorage accounts can only be used to store Azure file shares; no other storage resources (blob containers, queues, tables, etc.) can be deployed in a FileStorage account.
- Link : <https://docs.microsoft.com/en-us/azure/storage/files/storage-how-to-create-file-share?tabs=azure-portal>



# What is Azure Queue Storage

- Azure Queue Storage is a service for storing large numbers of messages. You access messages from anywhere in the world via authenticated calls using HTTP or HTTPS. A queue message can be up to 64 KB in size. A queue may contain millions of messages, up to the total capacity limit of a storage account. Queues are commonly used to create a backlog of work to process asynchronously.
- **Queue Storage concepts** : Queue Storage contains the following components



➤ **URL format:** Queues are addressable using the following URL format:

❑ `https://<storage account>.queue.core.windows.net/<queue>`

➤ **The following URL addresses a queue in the diagram:**

❑ <https://myaccount.queue.core.windows.net/images-to-download>

➤ **Storage account:** All access to Azure Storage is done through a storage account.

➤ **Queue:** A queue contains a set of messages. The queue name must be all lowercase. For information on naming queues, see Naming queues and metadata.

➤ **Message:** A message, in any format, of up to 64 KB. Before version 2017-07-29, the maximum time-to-live allowed is seven days. For version 2017-07-29 or later, the maximum time-to-live can be any positive number, or -1 indicating that the message doesn't expire. If this parameter is omitted, the default time-to-live is seven days.

# DEMO : Create Azure Queue Storage

- How to use Azure Queue Storage from Python :
- Link : <https://docs.microsoft.com/en-us/azure/storage/queues/storage-python-how-to-use-queue-storage?tabs=python2%2Cenvironment-variable-linux>



# Introduction of Azure Tables Storage

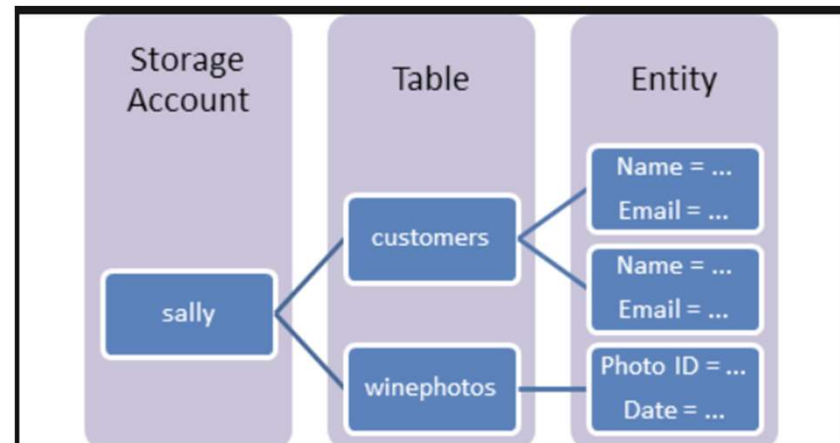
- **What is Azure Table storage :** Azure Table storage is a service that stores non-relational structured data (also known as structured NoSQL data) in the cloud, providing a key/attribute store with a schemaless design. Because Table storage is schemaless, it's easy to adapt your data as the needs of your application evolve. Access to Table storage data is fast and cost-effective for many types of applications and is typically lower in cost than traditional SQL for similar volumes of data.
- You can use Table storage to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires. You can store any number of entities in a table, and a storage account may contain any number of tables, up to the capacity limit of the storage account.

# What is Table storage

- **Azure Table storage** stores large amounts of structured data. The service is a NoSQL datastore which accepts **authenticated calls from inside and outside the Azure cloud**. Azure tables are ideal for storing structured, non-relational data. **Common uses of Table storage include:**
  - Storing TBs of structured data capable of serving web scale applications
  - Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
  - Quickly querying data using a clustered index
  - Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries
  - You can use Table storage to store and query huge sets of structured, non-relational data, and your tables will scale as demand increases.

# Table storage concepts

- Table storage contains the following components:
- **Table:** A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties.
- **Entity:** An entity is a set of properties, similar to a database row. An entity in Azure Storage can be up to 1MB in size. An entity in Azure Cosmos DB can be up to 2MB in size.
- **Properties:** A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has three system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.



URL format: Azure Table Storage accounts use this format:



`http://<storage account>.table.core.windows.net/<table>`



Azure Cosmos DB Table API accounts use this format:



`http://<storage account>.table.cosmosdb.azure.com/<table>`

## Azure Storage Account



entities

entities

entities

entities

- Doesn't enforce a schema on the entities within it

- A single table can, and often does, contain all kinds of entities with many different sets of properties

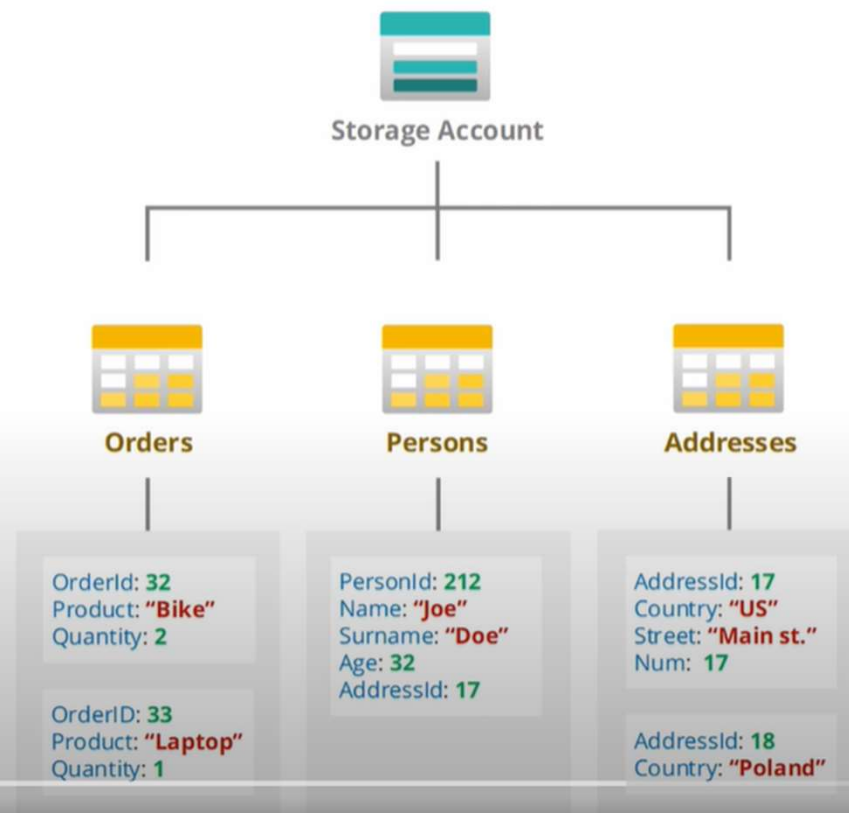
# How Azure Table Storage works?

## Storage Account

- Multi-purpose storage service (blob, file, queue and **table**)
- **Up to 20,000** rows/s (1KiB row)

## Table

- A collection of rows (**entity**)
- **Up to 500 TiB** of data
- **Up to 2,000** rows/s



## Entity

- Set of **properties** (columns and values)
- **Up to 1MB** of data
- **Max 255 properties**

## Property

- **Name-Value pair**
- Three system properties (PartitionKey, RowKey and Timestamp)

## DEMO : Create Azure Table Storage and Add Data in Table

- Create a Storage Table : Table Name as “Azure-Test-Storage-Table”
- Add Entities
- Add Properties like Key Value Pairs
- This can be done in Azure Storage Explorer only
- From Portal , only Table can be created .



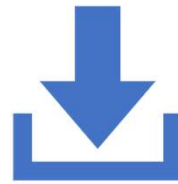
# What is Azure Storage Explorer

- Microsoft Azure Storage Explorer is a **standalone app that makes it easy to work with Azure Storage data on Windows, macOS, and Linux**. In this article, you'll learn several ways of connecting to and managing your Azure storage accounts
- The Azure Storage platform is **Microsoft's cloud storage solution for modern data storage scenarios**. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud

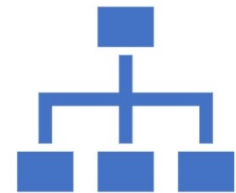


# How to Install Azure Storage Explorer and Set-up

---



Download and Install Azure Storage Explorer :  
<https://azure.microsoft.com/en-us/features/storage-explorer/>



Set-up Process :  
<https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer?tabs=windows#sign-in-to-azure>