**The lecture 13**

**Azure storage**

Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers a massively scalable object store for data objects, a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store. Azure Storage is:

* **Durable and highly available.** Redundancy ensures that your data is safe in the event of transient hardware failures. You can also opt to replicate data across datacenters 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.** All data written to Azure Storage 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.** Microsoft 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 services**

Azure Storage includes these data services:

* [Azure Blobs](https://docs.microsoft.com/en-gb/azure/storage/blobs/storage-blobs-introduction): A massively scalable object store for text and binary data.
* [Azure Files](https://docs.microsoft.com/en-gb/azure/storage/files/storage-files-introduction): Managed file shares for cloud or on-premises deployments.
* [Azure Queues](https://docs.microsoft.com/en-gb/azure/storage/queues/storage-queues-introduction): A messaging store for reliable messaging between application components.
* [Azure Tables](https://docs.microsoft.com/en-gb/azure/storage/tables/table-storage-overview): A NoSQL store for schemaless storage of structured data.

Each service is accessed through a storage account. To get started, see [Create a storage account](https://docs.microsoft.com/en-gb/azure/storage/common/storage-account-create).

**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, such as text or binary data.

Blob storage is ideal for:

* 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.

Objects in Blob storage can be accessed from anywhere in the world via HTTP or HTTPS. Users or client applications can access blobs via URLs, the [Azure Storage REST API](https://docs.microsoft.com/rest/api/storageservices/blob-service-rest-api), [Azure PowerShell](https://docs.microsoft.com/powershell/module/azure.storage), [Azure CLI](https://docs.microsoft.com/cli/azure/storage), or an Azure Storage client library. The storage client libraries are available for multiple languages, including [.NET](https://docs.microsoft.com/dotnet/api/overview/azure/storage/client), [Java](https://docs.microsoft.com/java/api/overview/azure/storage), [Node.js](https://azure.github.io/azure-storage-node), [Python](https://azure-storage.readthedocs.io/), [PHP](https://azure.github.io/azure-storage-php/), and [Ruby](https://azure.github.io/azure-storage-ruby).

For more information about Blob storage, see [Introduction to Blob storage](https://docs.microsoft.com/en-gb/azure/storage/blobs/storage-blobs-introduction).

**Azure Files**

[Azure Files](https://docs.microsoft.com/en-gb/azure/storage/files/storage-files-introduction) enables you to set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. That means that multiple VMs can share the same files with both read and write access. You can also read the files using the REST interface or the storage client libraries.

One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world using a URL that points to the file and includes a shared access signature (SAS) token. You can generate SAS tokens; they allow specific access to a private asset for a specific amount of time.

File shares can be used for many common scenarios:

* Many on-premises applications use file shares. This feature makes it easier to migrate those applications that share data to Azure. If you mount the file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal, if any, changes.
* Configuration files can be stored on a file share and accessed from multiple VMs. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.
* Diagnostic logs, metrics, and crash dumps are just three examples of data that can be written to a file share and processed or analyzed later.

At this time, Active Directory-based authentication and access control lists (ACLs) are not supported, but they will be at some time in the future. The storage account credentials are used to provide authentication for access to the file share. This means anybody with the share mounted will have full read/write access to the share.

For more information about Azure Files, see [Introduction to Azure Files](https://docs.microsoft.com/en-gb/azure/storage/files/storage-files-introduction).

**Queue storage**

The Azure Queue service is used to store and retrieve messages. Queue messages can be up to 64 KB in size, and a queue can contain millions of messages. Queues are generally used to store lists of messages to be processed asynchronously.

For example, say you want your customers to be able to upload pictures, and you want to create thumbnails for each picture. You could have your customer wait for you to create the thumbnails while uploading the pictures. An alternative would be to use a queue. When the customer finishes their upload, write a message to the queue. Then have an Azure Function retrieve the message from the queue and create the thumbnails. Each of the parts of this processing can be scaled separately, giving you more control when tuning it for your usage.

For more information about Azure Queues, see [Introduction to Queues](https://docs.microsoft.com/en-gb/azure/storage/queues/storage-queues-introduction).

**Table storage**

Azure Table storage is now part of Azure Cosmos DB. To see Azure Table storage documentation, see the [Azure Table Storage Overview](https://docs.microsoft.com/en-gb/azure/storage/tables/table-storage-overview). In addition to the existing Azure Table storage service, there is a new Azure Cosmos DB Table API offering that provides throughput-optimized tables, global distribution, and automatic secondary indexes. To learn more and try out the new premium experience, please check out [Azure Cosmos DB Table API](https://aka.ms/premiumtables).

For more information about Table storage, see [Overview of Azure Table storage](https://docs.microsoft.com/en-gb/azure/storage/tables/table-storage-overview).

**Disk storage**

An Azure managed disk is a virtual hard disk (VHD). You can think of it like a physical disk in an on-premises server but, virtualized. Azure managed disks are stored as page blobs, which are a random IO storage object in Azure. We call a managed disk ‘managed’ because it is an abstraction over page blobs, blob containers, and Azure storage accounts. With managed disks, all you have to do is provision the disk, and Azure takes care of the rest.

For more information about managed disks, see [Introduction to Azure managed disks](https://docs.microsoft.com/en-gb/azure/virtual-machines/windows/managed-disks-overview).

**Types of storage accounts**

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. Consider these differences before you create a storage account to determine the type of account that is best for your applications. The types of storage accounts are:

* **General-purpose v2 accounts**: Basic storage account type for blobs, files, queues, and tables. Recommended for most scenarios using Azure Storage.
* **General-purpose v1 accounts**: Legacy account type for blobs, files, queues, and tables. Use general-purpose v2 accounts instead when possible.
* **BlockBlobStorage accounts**: Storage accounts with premium performance characteristics for block blobs and append blobs. Recommended for scenarios with high transactions rates, or scenarios that use smaller objects or require consistently low storage latency.
* **FileStorage accounts**: Files-only storage accounts with premium performance characteristics. Recommended for enterprise or high performance scale applications.
* **BlobStorage accounts**: Legacy Blob-only storage accounts. Use general-purpose v2 accounts instead when possible.

The following table describes the types of storage accounts and their capabilities:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Storage account type** | **Supported services** | **Supported performance tiers** | **Supported access tiers** | **Replication options** | **Deployment model****1** | **Encryption****2** |
| General-purpose V2 | Blob, File, Queue, Table, Disk, and Data Lake Gen26 | Standard, Premium5 | Hot, Cool, Archive3 | LRS, GRS, RA-GRS, ZRS, GZRS (preview), RA-GZRS (preview)4 | Resource Manager | Encrypted |
| General-purpose V1 | Blob, File, Queue, Table, and Disk | Standard, Premium5 | N/A | LRS, GRS, RA-GRS | Resource Manager, Classic | Encrypted |
| BlockBlobStorage | Blob (block blobs and append blobs only) | Premium | N/A | LRS, ZRS4 | Resource Manager | Encrypted |
| FileStorage | File only | Premium | N/A | LRS, ZRS4 | Resource Manager | Encrypted |
| BlobStorage | Blob (block blobs and append blobs only) | Standard | Hot, Cool, Archive3 | LRS, GRS, RA-GRS | Resource Manager | Encrypted |

1Using the Azure Resource Manager deployment model is recommended. Storage accounts using the classic deployment model can still be created in some locations, and existing classic accounts continue to be supported. For more information, see [Azure Resource Manager vs. classic deployment: Understand deployment models and the state of your resources](https://docs.microsoft.com/azure/azure-resource-manager/resource-manager-deployment-model).

2All storage accounts are encrypted using Storage Service Encryption (SSE) for data at rest. For more information, see [Azure Storage Service Encryption for Data at Rest](https://docs.microsoft.com/azure/storage/common/storage-service-encryption).

3The Archive tier is available at level of an individual blob only, not at the storage account level. Only block blobs and append blobs can be archived. For more information, see [Azure Blob storage: Hot, Cool, and Archive storage tiers](https://docs.microsoft.com/azure/storage/blobs/storage-blob-storage-tiers).

4Zone-redundant storage (ZRS) and geo-zone-redundant storage (GZRS/RA-GZRS) (preview) are available only for standard General-purpose V2, BlockBlobStorage, and FileStorage accounts in certain regions. For more information about ZRS, see [Zone-redundant storage (ZRS): Highly available Azure Storage applications](https://docs.microsoft.com/azure/storage/common/storage-redundancy-zrs). For more information about GZRS/RA-GZRS, see [Geo-zone-redundant storage for highly availability and maximum durability (preview)](https://docs.microsoft.com/azure/storage/common/storage-redundancy-gzrs). For more information about other replication options, see [Azure Storage replication](https://docs.microsoft.com/azure/storage/common/storage-redundancy).

5Premium performance for general-purpose v2 and general-purpose v1 accounts is available for disk and page blob only. Premium performance for block or append blobs are only available on BlockBlobStorage accounts. Premium performance for files are only available on FileStorage accounts.

6Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob storage. Data Lake Storage Gen2 is only supported on General-purpose V2 storage accounts with Hierarchical namespace enabled. For more information on Data Lake Storage Gen2, see [Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/azure/storage/blobs/data-lake-storage-introduction).

For more information about storage account types, see [Azure storage account overview](https://docs.microsoft.com/en-gb/azure/storage/common/storage-account-overview).

**Securing access to storage accounts**

Every request to Azure Storage must be authorized. Azure Storage supports the following authorization methods:

* **Azure Active Directory (Azure AD) integration for blob and queue data.** Azure Storage supports authentication and authorization with Azure AD for the Blob and Queue services via role-based access control (RBAC). Authorizing requests with Azure AD is recommended for superior security and ease of use. For more information, see [Authorize access to Azure blobs and queues using Azure Active Directory](https://docs.microsoft.com/en-gb/azure/storage/common/storage-auth-aad).
* **Azure AD authorization over SMB for Azure Files (preview).** Azure Files supports identity-based authorization over SMB (Server Message Block) through Azure Active Directory Domain Services. Your domain-joined Windows virtual machines (VMs) can access Azure file shares using Azure AD credentials. For more information, see [Overview of Azure Active Directory authorization over SMB for Azure Files (preview)](https://docs.microsoft.com/en-gb/azure/storage/files/storage-files-active-directory-overview).
* **Authorization with Shared Key.** The Azure Storage Blob, Queue, and Table services and Azure Files support authorization with Shared Key.A client using Shared Key authorization passes a header with every request that is signed using the storage account access key. For more information, see [Authorize with Shared Key](https://docs.microsoft.com/rest/api/storageservices/authorize-with-shared-key).
* **Authorization using shared access signatures (SAS).** A shared access signature (SAS) is a string containing a security token that can be appended to the URI for a storage resource. The security token encapsulates constraints such as permissions and the interval of access. For more information, refer to [Using Shared Access Signatures (SAS)](https://docs.microsoft.com/en-gb/azure/storage/common/storage-sas-overview).
* **Anonymous access to containers and blobs.** A container and its blobs may be publicly available. When you specify that a container or blob is public, anyone can read it anonymously; no authentication is required. For more information, see [Manage anonymous read access to containers and blobs](https://docs.microsoft.com/en-gb/azure/storage/blobs/storage-manage-access-to-resources)

**Encryption**

There are two basic kinds of encryption available for the Storage services. For more information about security and encryption, see the [Azure Storage security guide](https://docs.microsoft.com/en-gb/azure/storage/blobs/security-recommendations).

**Encryption at rest**

Azure Storage encryption protects and safeguards your data to meet your organizational security and compliance commitments. Azure Storage automatically encrypts all data prior to persisting to the storage account and decrypts it prior to retrieval. The encryption, decryption, and key management processes are totally transparent to users. Customers can also choose to manage their own keys using Azure Key Vault. For more information, see [Azure Storage encryption for data at rest](https://docs.microsoft.com/en-gb/azure/storage/common/storage-service-encryption).

**Client-side encryption**

The Azure Storage client libraries provide methods for encrypting data from the client library before sending it across the wire and decrypting the response. Data encrypted via client-side encryption is also encrypted at rest by Azure Storage. For more information about client-side encryption, see [Client-side encryption with .NET for Azure Storage](https://docs.microsoft.com/en-gb/azure/storage/common/storage-client-side-encryption).

**Redundancy**

In order to ensure that your data is durable, Azure Storage replicates multiple copies of your data. When you set up your storage account, you select a redundancy option.

Replication options for a storage account include:

* [Locally-redundant storage (LRS)](https://docs.microsoft.com/en-gb/azure/storage/common/storage-redundancy-lrs): A simple, low-cost replication strategy. Data is replicated synchronously three times within the primary region.
* [Zone-redundant storage (ZRS)](https://docs.microsoft.com/en-gb/azure/storage/common/storage-redundancy-zrs): Replication for scenarios requiring high availability. Data is replicated synchronously across three Azure availability zones in the primary region.
* [Geo-redundant storage (GRS)](https://docs.microsoft.com/en-gb/azure/storage/common/storage-redundancy-grs): Cross-regional replication to protect against regional outages. Data is replicated synchronously three times in the primary region, then replicated asynchronously to the secondary region. For read access to data in the secondary region, enable read-access geo-redundant storage (RA-GRS).
* [Geo-zone-redundant storage (GZRS) (preview)](https://docs.microsoft.com/en-gb/azure/storage/common/storage-redundancy-gzrs): Replication for scenarios requiring both high availability and maximum durability. Data is replicated synchronously across three Azure availability zones in the primary region, then replicated asynchronously to the secondary region. For read access to data in the secondary region, enable read-access geo-zone-redundant storage (RA-GZRS).

For more information about disaster recovery, see [Disaster recovery and storage account failover (preview) in Azure Storage](https://docs.microsoft.com/en-gb/azure/storage/common/storage-disaster-recovery-guidance).

**Transferring data to and from Azure Storage**

You have several options for moving data into or out of Azure Storage. Which option you choose depends on the size of your dataset and your network bandwidth. For more information, see [Choose an Azure solution for data transfer](https://docs.microsoft.com/en-gb/azure/storage/common/storage-choose-data-transfer-solution).

**Pricing**

For detailed information about pricing for Azure Storage, see the [Pricing page](https://azure.microsoft.com/pricing/details/storage/blobs/).