100 XP

Introduction

3 minutes

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data.

After completing this module, you'll be able to:

- Identify the different types of storage accounts and the resource hierarchy for blob storage.
- Explain how data is securely stored and protected through redundancy.
- Create a block blob storage account by using the Azure Cloud Shell.

Next unit: Explore Azure Blob storage

Continue

How are we doing? ☆ ☆ ☆ ☆ ☆

Explore Azure Blob storage

3 minutes

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 does not 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.

Users or client applications can access objects in Blob storage via HTTP/HTTPS, from anywhere in the world. Objects in Blob storage are accessible via the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library.

An Azure Storage account is the top-level container for all of your Azure Blob storage. The storage account provides a unique namespace for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS.

Types of storage accounts

Azure Storage offers two performance levels of storage accounts, standard and premium. Each performance level supports different features and has its own pricing model.

- **Standard:** This is the standard general-purpose v2 account and is recommended for most scenarios using Azure Storage.
- **Premium:** Premium accounts offer higher performance by using solid-state drives. If you create a premium account you can choose between three account types, block blobs, page blobs, or file shares.

The following table describes the types of storage accounts recommended by Microsoft for most scenarios using Blob storage.

Storage account type	Supported storage services	Usage
Standard general-purpose v2	Blob, Queue, and Table storage, Azure Files	Standard storage account type for blobs, file shares, queues, and tables. Recommended for most scenarios using Azure Storage. If you want support for NFS file shares in Azure Files, use the premium file shares account type.
Premium block blobs	Blob storage	Premium storage account type 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.
Premium page blobs	Page blobs only	Premium storage account type for page blobs only.
Premium file shares	Azure Files	Premium storage account type for file shares only.

Access tiers for block blob data

Azure Storage provides different options for accessing block blob data based on usage patterns. Each access tier in Azure Storage is optimized for a particular pattern of data usage. By selecting the right access tier for your needs, you can store your block blob data in the most cost-effective manner.

The available access tiers are:

- The **Hot** access tier, which is optimized for frequent access of objects in the storage account. The Hot tier has the highest storage costs, but the lowest access costs. New storage accounts are created in the hot tier by default.
- The **Cool** access tier, which is optimized for storing large amounts of data that is infrequently accessed and stored for at least 30 days. The Cool tier has lower storage costs

and higher access costs compared to the Hot tier.

- The **Archive** tier, which is available only for individual block blobs. The archive tier is optimized for data that can tolerate several hours of retrieval latency and will remain in the Archive tier for at least 180 days. The archive 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.

If there is a change in the usage pattern of your data, you can switch between these access tiers at any time.

Next unit: Discover Azure Blob storage resource types

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Discover Azure Blob storage resource types

3 minutes

Blob storage offers three types of resources:

- The **storage account**.
- A **container** in the storage account
- A **blob** in a container

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 account name. The combination of the account name and the Azure Storage blob 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:

Copy

```
http://mystorageaccount.blob.core.windows.net
```

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. The container name must be lowercase.

Blobs

Azure Storage supports three types of blobs:

- **Block blobs** store text and binary data, up to about 4.7 TB. Block blobs are made up of blocks of data that can be managed individually.
- **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 TB in size. Page blobs store virtual hard drive (VHD) files and serve as disks for Azure virtual machines.

Next unit: Explore Azure Storage security features

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Explore Azure Storage security features

3 minutes

Azure Storage provides a comprehensive set of security capabilities that together enable developers to build secure applications:

- All data (including metadata) written to Azure Storage is automatically encrypted using Storage Service Encryption (SSE).
- Azure Active Directory (Azure AD) and Role-Based Access Control (RBAC) are supported for Azure Storage for both resource management operations and data operations, as follows:
  - You can assign RBAC roles scoped to the storage account to security principals and use Azure AD to authorize resource management operations such as key management.
  - Azure AD integration is supported for blob and queue data operations. You can assign RBAC roles scoped to a subscription, resource group, storage account, or an individual container or queue to a security principal or a managed identity for Azure resources.
- Data can be secured in transit between an application and Azure by using Client-Side Encryption, HTTPS, or SMB 3.0.
- OS and data disks used by Azure virtual machines can be encrypted using Azure Disk Encryption.
- Delegated access to the data objects in Azure Storage can be granted using a shared access signature.

Azure Storage encryption for data at rest

Azure Storage automatically encrypts your data when persisting it to the cloud. Encryption protects your data and help you meet your organizational security and compliance commitments. Data in Azure Storage is encrypted and decrypted transparently using 256-bit AES encryption, one of the strongest block ciphers available, and is FIPS 140-2 compliant. Azure Storage encryption is similar to BitLocker encryption on Windows.

Azure Storage encryption is enabled for all new and existing storage accounts and cannot be disabled. Because your data is secured by default, you don't need to modify your code or applications to take advantage of Azure Storage encryption.

Storage accounts are encrypted regardless of their performance tier (standard or premium) or deployment model (Azure Resource Manager or classic). All Azure Storage redundancy options support encryption, and all copies of a storage account are encrypted. All Azure Storage resources are encrypted, including blobs, disks, files, queues, and tables. All object metadata is also encrypted.

Encryption does not affect Azure Storage performance. There is no additional cost for Azure Storage encryption.

Encryption key management

You can rely on Microsoft-managed keys for the encryption of your storage account, or you can manage encryption with your own keys. If you choose to manage encryption with your own keys, you have two options:

- You can specify a *customer-managed* key to use for encrypting and decrypting all data in the storage account. A customer-managed key is used to encrypt all data in all services in your storage account.
- You can specify a *customer-provided* key on Blob storage operations. A client making a read or write request against Blob storage can include an encryption key on the request for granular control over how blob data is encrypted and decrypted.

The following table compares key management options for Azure Storage encryption.

	Microsoft-managed keys	**Customer-managed keys**	**Customer-provided keys**
Encryption/decryption operations	Azure	Azure	Azure
Azure Storage services supported	All	Blob storage, Azure Files	Blob storage
Key storage	Microsoft key store	Azure Key Vault	Azure Key Vault or any other key store
Key rotation responsibility	Microsoft	Customer	Customer

	Microsoft-managed keys	Customer-managed keys	Customer-provided keys
Key usage	Microsoft	Azure portal, Storage Resource Provider REST API, Azure Storage management libraries, PowerShell, CLI	Azure Storage REST API (Blob storage), Azure Storage client libraries
Key access	Microsoft only	Microsoft, Customer	Customer only

Next unit: Evaluate Azure Storage redundancy options

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Evaluate Azure Storage redundancy options

3 minutes

Azure Storage always stores multiple copies of your data so that it is protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.

When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability. The factors that help determine which redundancy option you should choose include:

- How your data is replicated in the primary region
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters
- Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable for any reason

Redundancy in the primary region

Data in an Azure Storage account is always replicated three times in the primary region. Azure Storage offers two options for how your data is replicated in the primary region.

- **Locally redundant storage (LRS)**: Copies your data synchronously three times within a single physical location in the primary region. LRS is the least expensive replication option, but is not recommended for applications requiring high availability or durability.
- **Zone-redundant storage (ZRS)**: Copies your data synchronously across three Azure availability zones in the primary region. For applications requiring high availability, Microsoft recommends using ZRS in the primary region, and also replicating to a secondary region.

Redundancy in a secondary region

For applications requiring high durability, you can choose to additionally copy the data in your storage account to a secondary region that is hundreds of miles away from the primary region. If your storage account is copied to a secondary region, then your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable.

When you create a storage account, you select the primary region for the account. The paired secondary region is determined based on the primary region, and can't be changed.

Azure Storage offers two options for copying your data to a secondary region:

- **Geo-redundant storage (GRS)** copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.
- **Geo-zone-redundant storage (GZRS)** copies your data synchronously across three Azure availability zones in the primary region using ZRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.

Next unit: Exercise: Create a block blob storage account

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Exercise: Create a block blob storage account

10 minutes

The block blob storage account type lets you create block blobs with premium performance characteristics. This type of storage account is optimized for workloads with high transactions rates or that require very fast access times.

In this exercise you will create a block blob storage account by using the Azure portal, and in the Cloud Shell using the Azure CLI.

Prerequisites

Before you begin make sure you have the following requirements in place:

- An Azure account with an active subscription. If you don't already have one, you can sign up for a free trial at https://azure.com/free .

Create account in the Azure portal

To create a block blob storage account in the Azure portal, follow these steps:

1. In the Azure portal, select **All services** > the **Storage** category > **Storage accounts**.
2. Under **Storage accounts**, select + **Create**.
3. In the **Subscription** field, select the subscription in which to create the storage account.
4. In the **Resource group** field, select **Create new** and enter *az204-blob-rg* as the name for the new resource group.
5. In the **Storage account name** field, enter a name for the account. Note the following guidelines:

- The name must be unique across Azure.
- The name must be between three and 24 characters long.
- The name can include only numbers and lowercase letters.

6. In the **Location** field, select a location for the storage account, or use the default location.

7. For the rest of the settings, configure the following:

Field	Value
Performance	Select **Premium**.
Premium account type	Select **Block blobs**.
Replication	Leave the default setting of **Locally-redundant storage (LRS)**.

8. Select **Review + create** to review the storage account settings.

9. Select **Create**.

Create account by using Azure Cloud Shell

1. Login to the Azure portal and open the Cloud Shell.

   - You can also login to the Azure Cloud Shell directly.

2. Create a new resource group. Replace `<myLocation>` with a region near you.

> **Note**
>
> Skip this step if you created a resource group in the *Create account in the Azure portal* section above.

Bash Copy

```
az group create --name az204-blob-rg --location <myLocation>
```

3. Create the block blob storage account. See Step 5 in the *Create account in the Azure portal* instructions above for the storage account name requirements. Replace `<myLocation>` with a region near you.

Bash Copy

```
az storage account create --resource-group az204-blob-rg --name \
<myStorageAcct> --location <myLocation> \
--kind BlockBlobStorage --sku Premium_LRS
```

Clean up resources

When you no longer need the resources in this walkthrough use the following command to delete the resource group and associated resources.

Bash Copy

```
az group delete --name az204-blob-rg --no-wait
```

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Knowledge check

3 minutes

Check your knowledge

1. Which of the following types of blobs are used to store virtual hard drive files?

- ○ Block blobs
- ○ Append blobs
- ◉ Page blobs

✔ That's correct. Page blobs store random access files up to 8 TB in size, and are used to store virtual hard drive (VHD) files and serve as disks for Azure virtual machines.

2. Which of the following types of storage accounts is recommended for most scenarios using Azure Storage?

- ○ General-purpose v2

✔ That's correct. This supports blobs, files, queues, and tables. It is recommended for most scenarios using Azure Storage.

- ◉ General-purpose v1

✘ That's incorrect. This is a legacy account type.

- ○ FileStorage

Next unit: Summary

Continue

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Summary

3 minutes

In this module you learned how to:

- Identify the different types of storage accounts and the resource hierarchy for blob storage.
- Explain how data is securely stored and protected through redundancy.
- Create a block blob storage account by using the Azure Cloud Shell.

Module complete:

Unlock achievement

How are we doing? ☆ ☆ ☆ ☆ ☆