**Create an Azure Storage account**

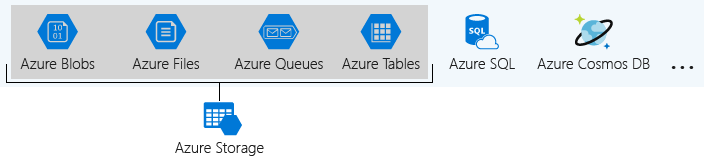
**Decide how many storage accounts you need**

Organizations often have multiple storage accounts to let them implement different sets of requirements. In the chocolate-manufacturer example, there would be one storage account for the private business data and one for the consumer-facing files. Here, you will learn the policy factors that are controlled by a storage account, which will help you decide how many accounts you need.

**What is Azure Storage?**

Azure provides many ways to store your data. There are multiple database options like Azure SQL Server, Azure Cosmos DB, and Azure Table Storage. Azure offers multiple ways to store and send messages, such as Azure Queues and Event Hubs. You can even store loose files using services like Azure Files and Azure Blobs.

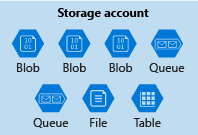
Azure selected four of these data services and placed them together under the name *Azure Storage*. The four services are Azure Blobs, Azure Files, Azure Queues, and Azure Tables. The following illustration shows the elements of Azure Storage.



These four were given special treatment because they are all primitive, cloud-based storage services and are often used together in the same application.

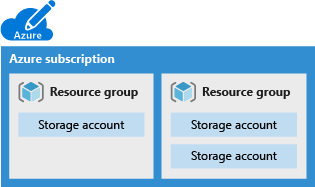
**What is a storage account?**

A *storage account* is a container that groups a set of Azure Storage services together. Only data services from Azure Storage can be included in a storage account (Azure Blobs, Azure Files, Azure Queues, and Azure Tables). The following illustration shows a storage account containing several data services.

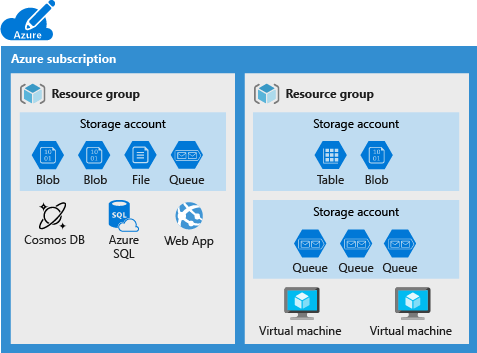


Combining data services into a storage account lets you manage them as a group. The settings you specify when you create the account, or any that you change after creation, are applied to everything in the account. Deleting the storage account deletes all of the data stored inside it.

A storage account is an Azure resource and is included in a resource group. The following illustration shows an Azure subscription containing multiple resource groups, where each group contains one or more storage accounts.



Other Azure data services like Azure SQL and Azure Cosmos DB are managed as independent Azure resources and cannot be included in a storage account. The following illustration shows a typical arrangement: Blobs, Files, Queues, and Tables are inside storage accounts, while other services are not.



**Storage account settings**

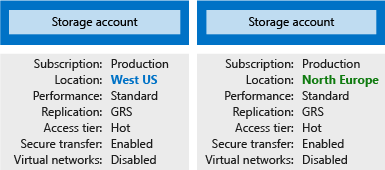
A storage account defines a policy that applies to all the storage services in the account. For example, you could specify that all the contained services will be stored in the West US datacenter, accessible only over https, and billed to the sales department's subscription.

The settings that are controlled by a storage account are:

* **Subscription**: The Azure subscription that will be billed for the services in the account.
* **Location**: The datacenter that will store the services in the account.
* **Performance**: Determines the data services you can have in your storage account and the type of hardware disks used to store the data. **Standard** allows you to have any data service (Blob, File, Queue, Table) and uses magnetic disk drives. **Premium** introduces additional services for storing data. For example, storing unstructured object data as block blobs or append blobs, and specialized file storage used to store and create premium file shares. These storage accounts use solid-state drives (SSD) for storage.
* **Replication**: Determines the strategy used to make copies of your data to protect against hardware failure or natural disaster. At a minimum, Azure will automatically maintain three copies of your data within the data center associated with the storage account. This is called locally-redundant storage (LRS), and guards against hardware failure but does not protect you from an event that incapacitates the entire datacenter. You can upgrade to one of the other options such as geo-redundant storage (GRS) to get replication at different datacenters across the world.
* **Access tier**: Controls how quickly you will be able to access the blobs in this storage account. Hot gives quicker access than Cool, but at increased cost. This applies only to blobs, and serves as the default value for new blobs.
* **Secure transfer required**: A security feature that determines the supported protocols for access. Enabled requires HTTPs, while disabled allows HTTP.
* **Virtual networks**: A security feature that allows inbound access requests only from the virtual network(s) you specify.

**How many storage accounts do you need?**

A storage account represents a collection of settings like location, replication strategy, and subscription owner. You need one storage account for every group of settings that you want to apply to your data. The following illustration shows two storage accounts that differ in one setting; that one difference is enough to require separate storage accounts.



The number of storage accounts you need is typically determined by your data diversity, cost sensitivity, and tolerance for management overhead.

**Data diversity**

Organizations often generate data that differs in where it is consumed, how sensitive it is, which group pays the bills, etc. Diversity along any of these vectors can lead to multiple storage accounts. Let's consider two examples:

1. Do you have data that is specific to a country or region? If so, you might want to locate it in a data center in that country for performance or compliance reasons. You will need one storage account for each location.
2. Do you have some data that is proprietary and some for public consumption? If so, you could enable virtual networks for the proprietary data and not for the public data. This will also require separate storage accounts.

In general, increased diversity means an increased number of storage accounts.

**Cost sensitivity**

A storage account by itself has no financial cost; however, the settings you choose for the account do influence the cost of services in the account. Geo-redundant storage costs more than locally-redundant storage. Premium performance and the Hot access tier increase the cost of blobs.

You can use multiple storage accounts to reduce costs. For example, you could partition your data into critical and non-critical categories. You could place your critical data into a storage account with geo-redundant storage and put your non-critical data in a different storage account with locally-redundant storage.

**Tolerance for management overhead**

Each storage account requires some time and attention from an administrator to create and maintain. It also increases complexity for anyone who adds data to your cloud storage; everyone in this role needs to understand the purpose of each storage account so they add new data to the correct account.

Storage accounts are a powerful tool to help you get the performance and security you need while minimizing costs. A typical strategy is to start with an analysis of your data and create partitions that share characteristics like location, billing, and replication strategy, and then create one storage account for each partition.

**Choose your account settings**

The storage account settings we've already covered apply to the data services in the account. Here, we will discuss the three settings that apply to the account itself, rather than to the data stored in the account:

* Name
* Deployment model
* Account kind

These settings impact how you manage your account and the cost of the services within it.

**Name**

Each storage account has a name. The name must be globally unique within Azure, use only lowercase letters and digits and be between 3 and 24 characters.

**Deployment model**

A *deployment model* is the system Azure uses to organize your resources. The model defines the API that you use to create, configure, and manage those resources. Azure provides two deployment models:

* **Resource Manager**: the current model that uses the Azure Resource Manager API
* **Classic**: a legacy offering that uses the Azure Service Management API

Most Azure resources only work with Resource Manager, and makes it easy to decide which model to choose. However, storage accounts, virtual machines, and virtual networks support both, so you must choose one or the other when you create your storage account.

The key feature difference between the two models is their support for grouping. The Resource Manager model adds the concept of a *resource group*, which is not available in the classic model. A resource group lets you deploy and manage a collection of resources as a single unit.

Microsoft recommends that you use **Resource Manager** for all new resources.

**Account kind**

Storage account *kind* is a set of policies that determine which data services you can include in the account and the pricing of those services. There are three kinds of storage accounts:

* **StorageV2 (general purpose v2)**: the current offering that supports all storage types and all of the latest features
* **Storage (general purpose v1)**: a legacy kind that supports all storage types but may not support all features
* **Blob storage**: a legacy kind that allows only block blobs and append blobs

Microsoft recommends that you use the **General-purpose v2** option for new storage accounts.

There are a few special cases that can be exceptions to this rule. For example, pricing for transactions is lower in general purpose v1, which would allow you to slightly reduce costs if that matches your typical workload.

The core advice here is to choose the **Resource Manager** deployment model and the **StorageV2 (general purpose v2)** account kind for all your storage accounts. The other options still exist primarily to allow existing resources to continue operation. For new resources, there are few reasons to consider the other choices.

Click +New, then click Storage account:

