



# Simple Storage Service

Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers.

Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

## Advantages of using Amazon S3

Amazon S3 is intentionally built with a minimal feature set that focuses on simplicity and robustness. Following are some of the advantages of using Amazon S3:

- **Creating buckets** – Create and name a bucket that stores data. Buckets are the fundamental containers in Amazon S3 for data storage.
- **Storing data** – Store an infinite amount of data in a bucket. Upload as many objects as you like into an Amazon S3 bucket. Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer-assigned key.
- **Downloading data** – Download your data or enable others to do so. Download your data anytime you like, or allow others to do the same.
- **Permissions** – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms can help keep data secure from unauthorized access.
- **Standard interfaces** – Use standards-based REST and SOAP interfaces designed to work with any internet-development toolkit.



## Amazon S3 Concepts

### • Buckets

A bucket is a container for objects stored in Amazon S3. Every object is contained in a bucket. For example, if the object named **photos/puppy.jpg** is stored in the **awsexamplebucket1** bucket in the US West (Oregon) Region, then it is addressable using the URL <https://awsexamplebucket1.s3.us-west-2.amazonaws.com/photos/puppy.jpg>.

Buckets serve several purposes:

- They organize the Amazon S3 namespace at the highest level.
- They identify the account responsible for storage and data transfer charges.
- They play a role in access control.
- They serve as the unit of aggregation for usage reporting.

### • Objects

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. These include some default metadata, such as the date last modified, and standard HTTP metadata, such as **Content-Type**. You can also specify custom metadata at the time the object is stored.

### • Keys

A key is the unique identifier for an object within a bucket. Every object in a bucket has exactly one key. The combination of a bucket, key, and version ID uniquely identify each object. So you can think of Amazon S3 as a basic data map between "bucket + key + version" and the object itself. Every object in Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, key, and optionally, a version. For example, in the URL <https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>, "doc" is the name of the bucket and "2006-03-01/AmazonS3.wsdl" is the key.



- **Regions**

You can choose the geographical AWS Region where Amazon S3 will store the buckets that you create. You might choose a Region to optimize latency, minimize costs, or address regulatory requirements. Objects stored in a Region never leave the Region unless you explicitly transfer them to another Region. For example, objects stored in the Europe (Ireland) Region never leave it.

## Amazon S3 Features

- **Storage Classes**

Amazon S3 offers a range of storage classes designed for different use cases. These include Amazon S3 STANDARD for general-purpose storage of frequently accessed data, Amazon S3 STANDARD\_IA for long-lived, but less frequently accessed data, and S3 Glacier for long-term archive.

- **Bucket Policies**

Bucket policies provide centralized access control to buckets and objects based on a variety of conditions, including Amazon S3 operations, requesters, resources, and aspects of the request (for example, IP address). The policies are expressed in the *access policy language* and enable centralized management of permissions. The permissions attached to a bucket apply to all of the objects in that bucket.

Both individuals and companies can use bucket policies. When companies register with Amazon S3, they create an *account*. Thereafter, the company becomes synonymous with the account. Accounts are financially responsible for the AWS resources that they (and their employees) create. Accounts have the power to grant bucket policy permissions and assign employees permissions based on a variety of conditions. For example, an account could create a policy that gives a user write access:

- To a particular S3 bucket
- From an account's corporate network
- During business hours



- **AWS Identity and Access Management**

You can use AWS Identity and Access Management (IAM) to manage access to your Amazon S3 resources.

For example, you can use IAM with Amazon S3 to control the type of access a user or group of users has to specific parts of an Amazon S3 bucket your AWS account owns.

- **Access Control Lists**

You can control access to each of your buckets and objects using an access control list (ACL). Access control lists (ACLs) are one of the resource-based access policy options that you can use to manage access to your buckets and objects. You can use ACLs to grant basic read/write permissions to other AWS accounts.

- **Versioning**

You can use versioning to keep multiple versions of an object in the same bucket. Use Amazon S3 Versioning to keep multiple versions of an object in one bucket. For example, you could store **my-image.jpg** (version 111111) and **my-image.jpg** (version 222222) in a single bucket. S3 Versioning protects you from the consequences of unintended overwrites and deletions. You can also use it to archive objects so that you have access to previous versions.

- **Operations**

Following are the most common operations that you'll execute through the API.

Common operations:

- **Create a bucket** – Create and name your own bucket in which to store your objects.
- **Write an object** – Store data by creating or overwriting an object. When you write an object, you specify a unique key in the namespace of your bucket. This is also a good time to specify any access control you want on the object.
- **Read an object** – Read data back. You can download the data via HTTP or BitTorrent.
- **Delete an object** – Delete some of your data.
- **List keys** – List the keys contained in one of your buckets. You can filter the key list based on a prefix.



### Paying for Amazon S3

Pricing for Amazon S3 is designed so that you don't have to plan for the storage requirements of your application. Most storage providers force you to purchase a predetermined amount of storage and network transfer capacity: If you exceed that capacity, your service is shut off or you are charged high overage fees. If you do not exceed that capacity, you pay as though you used it all.

Amazon S3 charges you only for what you actually use, with no hidden fees and no overage charges. This gives developers a variable-cost service that can grow with their business while enjoying the cost advantages of the AWS infrastructure.

Before storing anything in Amazon S3, you must register with the service and provide a payment method that is charged at the end of each month. There are no setup fees to begin using the service. At the end of the month, your payment method is automatically charged for that month's usage.

## Hosting a Static Website

You can use Amazon S3 to host a static website. On a *static* website, individual webpages include static content. They might also contain client-side scripts.

By contrast, a *dynamic* website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET. Amazon S3 does not support server-side scripting, but AWS has other resources for hosting dynamic websites.

To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you enable static website hosting, set permissions, and add an index document. Depending on your website requirements, you can also configure other options, including redirects, web traffic logging, and custom error documents.

After you configure your bucket as a static website, you can access the bucket through the AWS Region-specific Amazon S3 website endpoints for your bucket.