AWS S3

## Service Overview. Concepts.

*Amazon Simple Storage Service (Amazon S3) is one of the main building blocks of AWS.*

*Amazon S3 is storage for the Internet. It is designed and advertised as ”infinitely scaling” storage. Many websites use Amazon S3 as a backbone.*

*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.*

## Amazon S3 concepts

*It is also necessary to draw your special attention to the fact that unlike many others store solutions in Amazon like EBS or Instance Storage which acts as network drive you can attach to your instances while they run, AWS S3 is an* ***object storage****. For an accurate and good explanation of the difference, see* [*AWS EBS and S3: Object Storage Vs. Block Storage in the AWS Cloud*](https://cloud.netapp.com/blog/block-storage-vs-object-storage-cloud)

## Buckets

*A Bucket is the fundamental container for objects stored in Amazon S3. Drawing parallels with the traditional filesystems, Amazon S3 allows to store objects (files) in “buckets” (directories).*

*Some facts about Buckets:*

* *Buckets must have a globally unique name*
* *Buckets are defined at the region level*

*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.*

*For more information about buckets, see* [*Buckets overview*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html)*.*

*To understand how to work with buckets, see* [*Creating, configuring, and working with Amazon S3 buckets*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-buckets-s3.html)*.*

## Objects

*Objects are the fundamental entities stored in Amazon S3.* *An object is uniquely identified within a bucket by a key (name) and a version ID.*

*An object consists of the following:*

1. *Key*
2. *Version ID*
3. *Value*
4. *Some Metadata*
5. *Subresources*
6. *Access control information*

*For more information about objects and working with it, see* [*Uploading, downloading, and working with objects in Amazon S3*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/uploading-downloading-objects.html)

## Regions

*Buckets tied to the 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.*

*For more information, see* [*Regions and Endpoints*](https://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region)*.*

## Amazon S3 features

## Storage Classes

*Each object in Amazon S3 has a storage class associated with it.*

*For example, if you list the objects in an S3 bucket, the console shows the storage class for all the objects in the list.*

*Amazon S3 offers a range of storage classes for the objects that you store. You choose a class depending on your use case scenario and performance access requirements. All of these storage classes offer high durability.*

***The main S3 storage classes*** *include (but are not limited to) the following:*

* ***S3 Standard*** *for general-purpose storage of frequently accessed data;*
* ***S3 Standard\_IA*** *for long-lived, but less frequently accessed data;*
* ***S3 Glacier*** *for long-term archive.*

*For a quick comparison of Storage Classes table, see* [*Comparing the Amazon S3 storage classes*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-class-intro.html#sc-compare)*.*

*For more information about Storage Classes, choosing the right storage class for you data and using it, see* [*Using Amazon S3 storage classes*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-class-intro.html)*.*

## S3 Security

*Security is a shared responsibility between AWS and you. The* [*shared responsibility model*](https://aws.amazon.com/compliance/shared-responsibility-model/) *describes this as security of the cloud and security in the cloud.*

*Security in AWS S3 is provided at different levels. S3 Bucket security can be:*

***User based***

* ***IAM policies****; controls which API calls should be allowed for a specific user from IAM console.*

***Resource Based***

* ***Bucket Policies*** *is bucket-wide rules from the* [*S3 console*](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_s3_rw-bucket.html) *(it also allows cross account access);*
* ***Object Access Control List (ACL****) provides finer grain control;*
* ***Bucket ACL*** *– less common.*

*To go straight to the S3 security tips and best practices, visit the following section of the Amazon documentation:* [*Security Best Practices for Amazon S3*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html#server-side)*.*

## 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.*

*For more information about IAM, see the following:*

* [*AWS Identity and Access Management (IAM)*](https://aws.amazon.com/iam/)
* [*Getting started*](https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started.html)

## [Bucket policies](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#BucketPolicies)

*Using* ***S3 bucket policy*** *may help to:*

* *Grant public access to the bucket*
* *Force objects to be encrypted at upload*
* *Grant access to another account (Cross Account)*

*For more information about Bucket policies and its usage, see* [*Bucket policies and user policies*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-iam-policies.html)*.*

## Access control lists

*You can control access to each of your buckets and objects using an access control list (ACL).*

*For more information, see* [*Access control list (ACL) overview*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html)*.*

## S3 Encryption for Objects

## Encryption at rest

***Server-side encryption*** *is the encryption of data at its destination by the application or service that receives it. Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it.*

*For more information, see* [*Protecting data using server-side encryption*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html)*.*

***Client-Side encryption*** *encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools. As with server-side encryption, client-side encryption can help reduce risk by encrypting the data with a key that is stored in a different mechanism than the mechanism that stores the data itself.*

*Amazon S3 provides multiple client-side encryption options. For more information, see* [*Protecting data using client-side encryption*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingClientSideEncryption.html)*.*

## Encryption in transit

*You can use HTTPS (TLS) to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. You should allow only encrypted connections over HTTPS (TLS) using the* [*aws:SecureTransport*](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html#Conditions_Boolean) *condition on Amazon S3 bucket policies.*

*Also consider implementing on-going detective controls using the* [*s3-bucket-ssl-requests-only*](https://docs.aws.amazon.com/config/latest/developerguide/s3-bucket-ssl-requests-only.html) *managed AWS Config rule.*

* [*IAM User Guide*](https://docs.aws.amazon.com/IAM/latest/UserGuide/)

## S3 Security - Other

*Other S3 securities you should know about:*

* *Networking:*
  + *You can access S3 privately through VPC Endpoints (for instances in VPC without www internet)*
* *Logging and Audit:*
  + *S3 Access Logs can be stored in other S3 bucket*
  + *API calls can be logged in AWS CloudTrail*
* *User Security:*
  + *MFA Delete: MFA (multi factor authentication) can be required in versioned buckets to delete objects,* [*see details*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/MultiFactorAuthenticationDelete.html)
  + *Pre-Signed URLs: URLs that are valid only for a limited time (example: premium video service for users who signed in and purchased this video),* [*see details*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html)

*For more information about S3 security, see* [*Amazon S3 Security*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/security.html)*.*

## Versioning

*AWS S3 supports* ***versioning****. You can use versioning to keep multiple versions of an object in the same S3 bucket.*

*Some facts about versioning:*

* *It is enabled at the bucket level*
* *Same key overwrite will increment the “version”: 1, 2, 3….*
* *It is best practice to version your buckets*
* *Versioning protects against unintended deletes (have ability to restore a version)*
* *Easy roll back to previous version*

*Any file that is not versioned prior to enabling versioning will have version “null”. Suspending versioning does not delete the previous versions.*

*For more information about versioning, see* [*Using Versioning*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html)*.*

## Operations

## Working with Amazon S3 buckets

*To store an object in Amazon S3, you create a bucket and then upload the object to a bucket. When the object is in the bucket, you can open it, download it, and move it. When you no longer need an object or a bucket, you can clean up your resources.*

*For more information about using S3 buckets, see the following links:*

* [*Creating a bucket*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html)
* [*Viewing the properties for an S3 bucket*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/view-bucket-properties.html)
* [*Accessing a bucket*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-bucket-intro.html)
* [*Emptying a bucket*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/empty-bucket.html)
* [*Deleting a bucket*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/delete-bucket.html)
* [*Setting default server-side encryption behavior for Amazon S3 buckets*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-encryption.html)
* [*Configuring fast, secure file transfers using Amazon S3 Transfer Acceleration*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/transfer-acceleration.html)
* [*Using Requester Pays buckets for storage transfers and usage*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/RequesterPaysBuckets.html)
* [*Bucket restrictions and limitations*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/BucketRestrictions.html)

## Amazon S3 performance

*Your applications can easily achieve thousands of transactions per second in request performance when uploading and retrieving storage from Amazon S3. Amazon S3 automatically scales to high request rates. For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per* [*prefix*](https://docs.aws.amazon.com/general/latest/gr/glos-chap.html#keyprefix) *in a bucket. There are no limits to the number of prefixes in a bucket.*

*You can increase your read or write performance by* ***parallelizing reads****. For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second. Similarly, you can scale write operations by writing to multiple prefixes.*

*Also, you can use* ***multi-part uploads*** *for big files (recommended >100MB and must use for >5GB),* [*see details*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/mpuoverview.html)*.*

*When downloading from S3, use* ***Byte-Range Fetches****,* [*see details*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance-guidelines.html#optimizing-performance-guidelines-get-range)

*If you use SSE-KMS, you may be impacted by the KMS limits*

*For more information about performance, see* [*Optimizing Amazon S3 performance*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html)*.*

## Use cases

*The following table summarizes some of the most common advanced functionality offered by Amazon S3. Note that some advanced functionality is not available in the AWS Management Console and requires that you use the Amazon S3 API.*

| **Link** | **Functionality** |
| --- | --- |
| [Using Requester Pays buckets for storage transfers and usage](https://docs.aws.amazon.com/AmazonS3/latest/userguide/RequesterPaysBuckets.html) | Learn how to configure a bucket so that a customer pays for the downloads they make. |
| [Publishing content using Amazon S3 and BitTorrent](https://docs.aws.amazon.com/AmazonS3/latest/userguide/S3Torrent.html#S3TorrentPublish) | Use BitTorrent, which is an open, peer-to-peer protocol for distributing files. |
| [Using versioning in S3 buckets](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html) | Learn about Amazon S3 versioning capabilities. |
| [Hosting a static website using Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html) | Learn how to host a static website on Amazon S3. |
| [Managing your storage lifecycle](https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html) | Learn how to manage the lifecycle of objects in your bucket. Lifecycle management includes expiring objects and archiving objects (transitioning objects to the S3 S3 Glacier storage class). |

*For more information, see* [*Amazon S3 use cases*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/S3-gsg-AdvancedAmazonS3Features.html)*.*

## Governance

*Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon S3 and your AWS solutions. For information about monitoring Amazon S3, see* [*Monitoring Amazon S3*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/monitoring-overview.html)*.*

*You can use analytics and insights in Amazon S3 to understand, analyze, and optimize your storage usage. For information about analytics and insights for Amazon S3, see* [*Using analytics and insights*](https://docs.aws.amazon.com/AmazonS3/latest/userguide/analytics-insights.html)*.*

## Cautions

*You can only access Amazon S3 and its features only in AWS Regions that are enabled for your account.*

## Pricing considerations

*For information about Amazon S3 prices, see* [*Amazon S3 pricing*](https://aws.amazon.com/s3/pricing/?nc=sn&loc=4)

## More details

*Use official Amazon documentation* [*https://docs.aws.amazon.com/s3/index.html*](https://docs.aws.amazon.com/s3/index.html)