Introduction to S3

Amazon Simple Storage Service (Amazon S3) is an object-based storage service that offers industry-leading scalability, data availability, security, and performance.

* This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.
* Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.
* Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

One of AWS's oldest services, Amazon S3 could be defined as AWS object-based file storage service.

* Amazon S3 stores data in buckets as objects.
* An object consists of a file or metadata that optionally identifies this file.

S3 – Bucket

* A bucket is a logical storage unit used to store objects in AWS.
* A bucket can also be considered as a container and also the first thing we create in S3.
* Stored objects in a bucket may be in the form of any files such as text, image, movie, video, etc.
* Objects are consist of keys and values. The key is the name of the object, and the value is the data that the object stores.
* An S3 bucket should be first created in one of the AWS Regions to upload data to Amazon S3. Then any number of objects can be added into that bucket.
* Folders like objects can also be created in a bucket.
* The number of objects that can be stored in a bucket is not limited, but each AWS account can only have 100 buckets at once.
* Since S3 is a global service, a region must be selected when creating a bucket. When you store anything in S3, it's replicated across AZs in that region.
* S3 also saves HTML files in buckets and lets us host static web pages without having to maintain a server.
* S3 is an object-based storage system, not a blocked-based. Buckets are not available for mounting into a server as a drive or a disk.

What is Object in S3 ?

* An object can be any kind of file; a text file, a photo, a video, etc.
* Objects are consist of keys and values.
* The key is the name of the object, and the value is the data that the object stores.
* Objects have metadata that means information about data. For example, the creation date of the object, size, content length variables are metadata that are kept with that object.
* An unlimited number of data objects can be added to a bucket.
* Objects exist in buckets that are created in a specific AWS Region. Objects never leave that region unless you explicitly transfer them to a different region.
* The object size of the data can be up to 5 TB. Max 5 TB size is a limit for a single file. It is unlimited in terms of the number of files you can put in S3. You can put as many files as you want.
* The max. size of an object you can upload via AWS Management Console is 160 GB. For uploading a file greater than 160 GB, the AWS CLI, AWS SDK, or Amazon S3 REST API is needed to be used.
* Objects can also be moved to a created folder in S3.
* If the object uploaded in a bucket no longer needs to be stored, it should be deleted to prevent further charges. Because while the bucket is free to create, objects uploaded into buckets are charged as long as they are stored in the bucket.
* Max 10 tags can be defined per object.

Storage Class in AWS

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. There are 8 types of storage class in AWS:

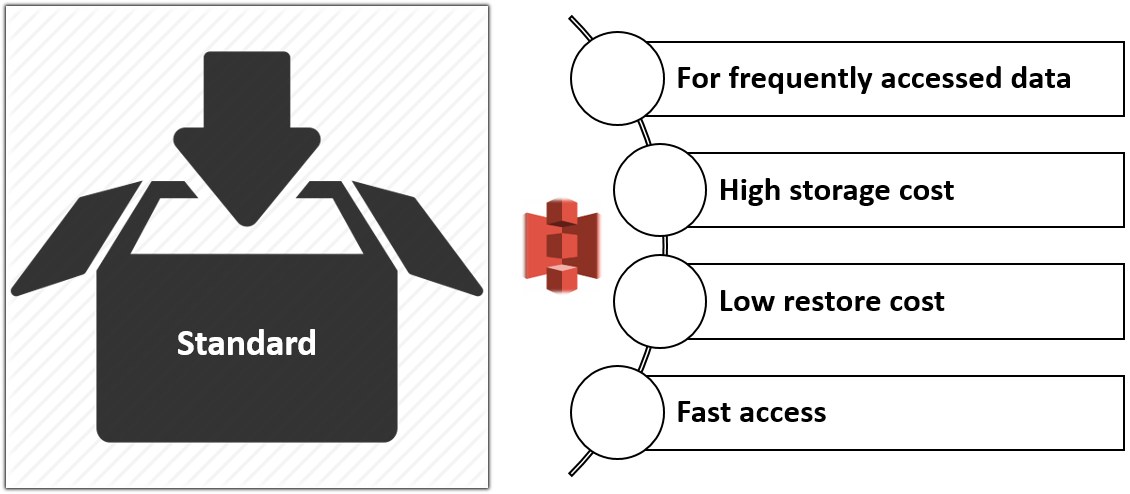
Standard Reduced Redundancy

Intelligent-Tiering Standart-IA

One Zone-IA Glacier

Glacier Deep Archive AWS S3 Outposts

### Standard Storage



Standard is the default storage class. If users don't specify the storage class when uploading an object, Amazon S3 assigns the STANDARD storage class. This is the basic storage solution for the objects that you always need and use.

Standard storage class has the following features:

* High capacity and low latency.
* Reliability at 99,999999999% level (Customers run the risk of losing only one of 100 billion objects per year).
* Availability at 99,99% level (For 10 thousand hours, the data will not be available only within one hour).

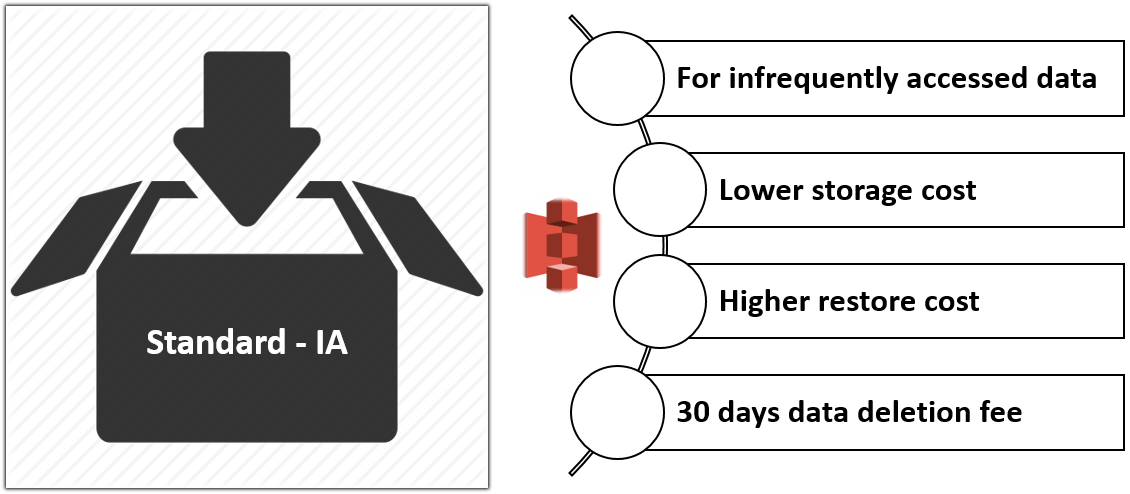
The scenarios such as cloud applications and web-services, mobile games and apps, website hosting, content distribution can be counted as the most appropriate areas of use for standard storage class.

### Reduced Redundancy Storage (RRS)

The Reduced Redundancy Storage (RRS) storage class is designed for noncritical, reproducible data that can be stored with less redundancy than the STANDARD storage class. It allows reducing the storage costs for uncritical data.

* The main difference between RRS and S3 Standard is reliability at 99,99%.
* This means that if you store 10,000 items, you run the risk of losing just one of them within a year.
* If an RRS object is lost, when requests are made to that object, Amazon S3 returns a 405 error.
* AWS recommends not to use this storage class because the Standard storage class is more cost-effective.

### Standard IA



Standard IA (Infrequent Access) is a convenient storage solution for files that are not frequently accessed but need to be accessed quickly when needed. It designed for the data which requires less frequent access, but with longer storage time than in case of Standard.

Standard IA storage class has the following features:

* Lower storage cost
* Higher restore cost
* Availability at 99.9% level within a year (For 10 thousand hours, the data will not be available within 10 hours).
* Data retrieval charge

If you access a file frequently, you should keep it in the normal standard layer. Because in Standard-IA Infrequent Access, extra money is deducted each time you access the files you keep, apart from the price you pay normally. So if an object is accessed every day and used frequently, it is cheaper to keep it as standard. But if the object is accessed 2 times a year, it is cheaper to transfer it to Standard-IA Infrequent Access and keep it there.

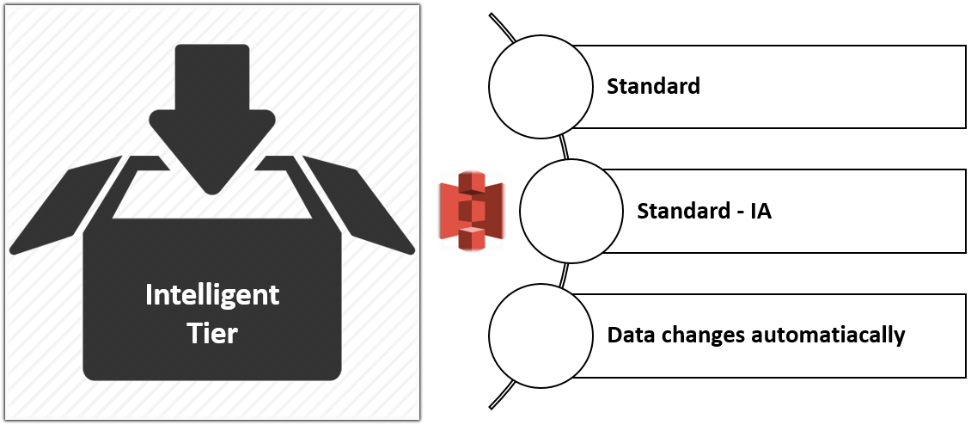
### One Zone IA

AWS introduced another Amazon S3 storage class in April 2018, One-Zone IA (Infrequent Access) which is 20 percent less expensive than Standard IA due to less availability. Instead of three available zones such as the other storage classes, One Zone IA only stores data in one.

One-Zone IA can be considered as an affordable alternative storage class for files that are rarely used and can be afforded to lose.

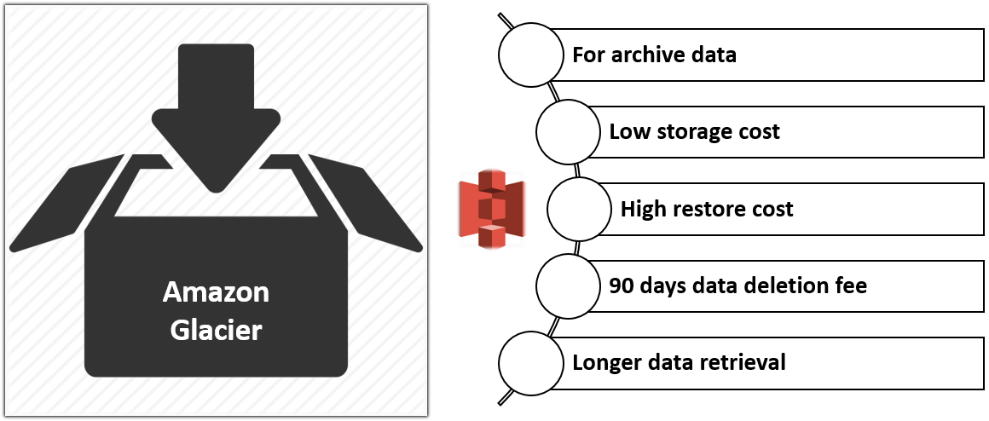
### Intelligent Tier

The **Intelligent Tier**storage class is designed to optimize storage costs by automatically moving data to the most cost-effective storage access tier, without performance impact or operational overhead.



* It delivers automatic cost savings by moving data on a granular object-level between two access tiers, a frequent access tier and a lower-cost infrequent access tier, when access patterns change.
* It is ideal if you want to optimize storage costs automatically for long-lived data when access patterns are unknown or unpredictable.
* For example, if an object has not been accessed in 30 days, AWS will move it to the infrequent access.
* When the object is then accessed after being transferred to infrequent access, AWS can move it back for cheaper subsequent accesses to the frequent access storage class.

### Amazon Glacier



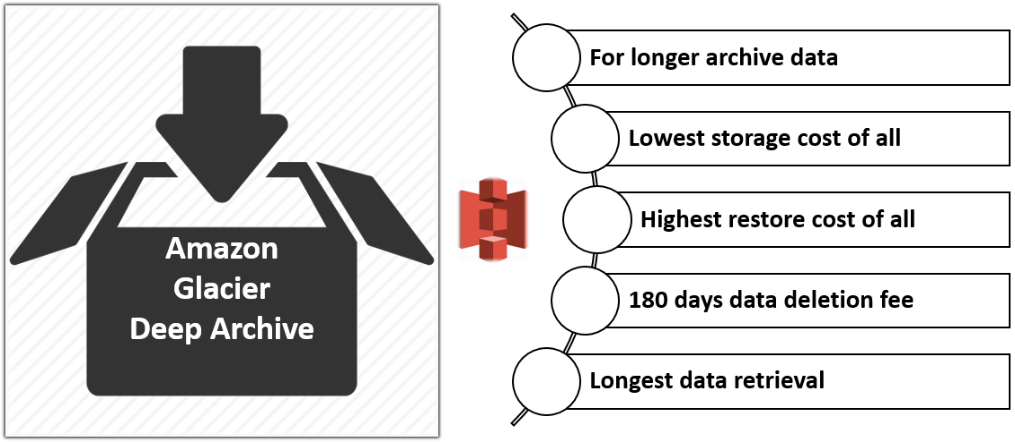
Amazon Glacier is an independent service from S3. It is a perfect solution for long-term storage and data archiving that doesn't require instant access.

* For example, you have backups you need once a year. These are not urgent, you don't have to access them in milliseconds when you need them.
* Here you can store such files at a very affordable price in Glacier service.
* Data stored in the GLACIER storage class has a minimum storage the duration period of 90 days and can be accessed in as little as 1-5 minutes using expedited retrieval.
* If you have deleted, overwritten, or transitioned to a different storage class an object before the 90-day minimum, you are charged for 90 days.

The following scenarios can be thought as the most appropriate examples of use for Glacier.

* Media resources archives
* Backup copies of databases with long storage.
* Archives of organizations working in the different businesses such as health, sport, insurance, etc.

### Amazon Glacier Deep Archive



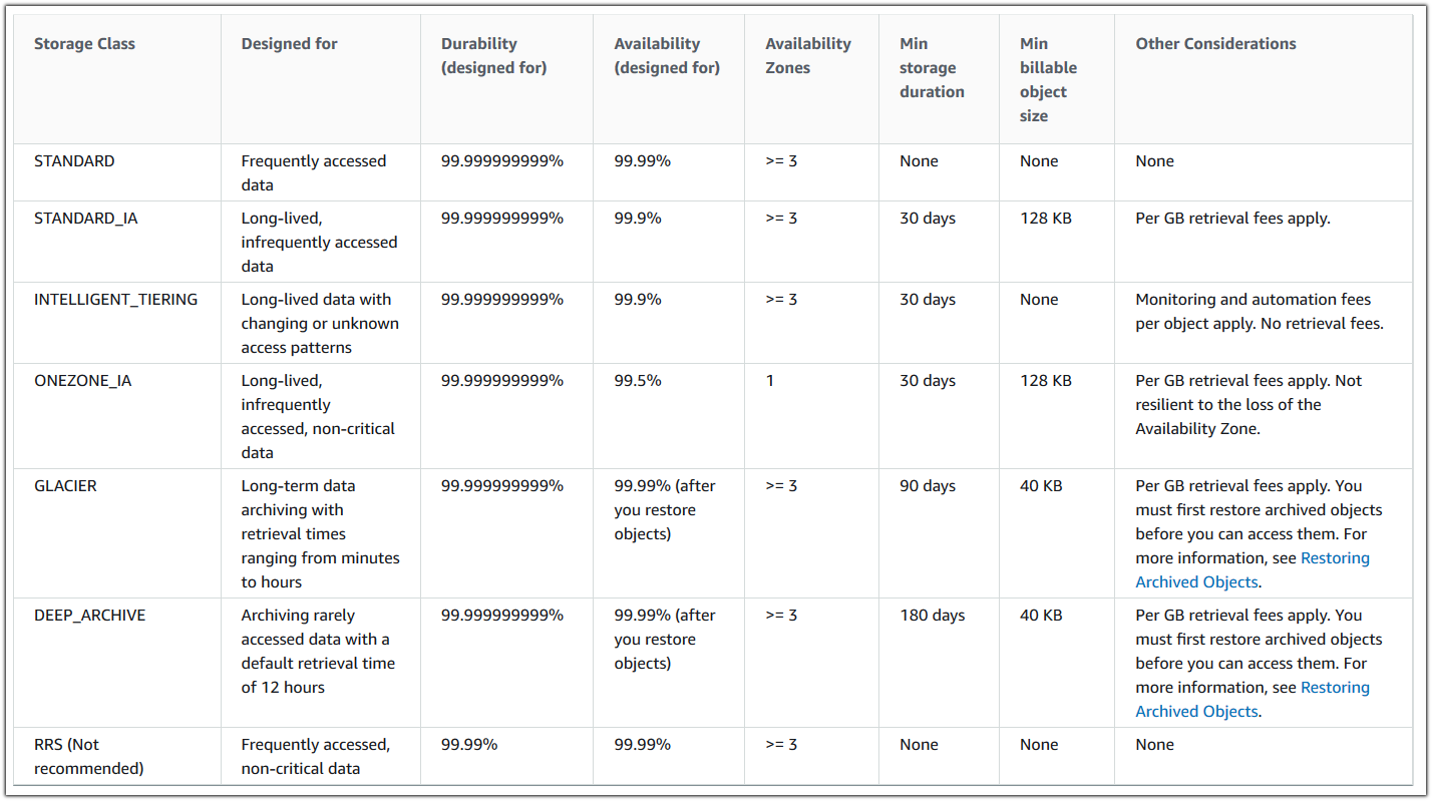
Amazon Glacier Deep Archive is the lowest cost storage option in AWS. Storage costs for DEEP\_ARCHIVE is less expensive than using the Glacier storage class.

* It is used for archiving data that rarely need to be accessed.
* You can reduce Amazon Glacier Deep Archive retrieval costs by using bulk retrieval, which returns data within 48 hours.
* The fastest retrieval time is up to 12 hours.
* Data stored in the Amazon Glacier Deep Archive storage class has a minimum storage duration period of 180 days and a default retrieval time of 12 hours.
* If you have deleted, overwritten, or transitioned to a different storage class an object before the 180-day minimum, you are charged for 180 days.

### AWS S3 Outposts

* AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to your premises. By providing local access to AWS-managed infrastructure, AWS Outposts helps you build and run applications on-premises using the same programming interfaces as in AWS Regions while using local compute and storage resources for lower latency and local data processing needs.
* With Amazon S3 on Outposts, you can create S3 buckets on your AWS Outposts and easily store and retrieve objects on-premises for applications that require local data access, local data processing, and data residency.
* S3 Outposts is a new storage class that uses the S3 APIs and is designed to store data durably and redundantly across multiple devices and servers on your AWS Outposts.
* So you can store your object in your on-premises data center with S3 capabilities including S3 outposts storages classes.

### Storage Classes Comparison



## S3 Versioning

**Versioning** is a way to keep multiple versions of an object in a bucket.

* It is used to **manage**, **delete**, and **restore** **any version of any object** saved in an S3 bucket.
* Each time an object in a bucket changed, a new version of the object would be created and act as a new current version.
* By using versioning, all unwanted user behaviour and program errors can be quickly recovered.

For example:

* If you delete an object, instead of removing it permanently, Amazon S3 inserts a delete marker, which becomes the current object version. You can always restore the previous version.
* If you overwrite an object, it results in a new object version in the bucket.

### States of Versioning

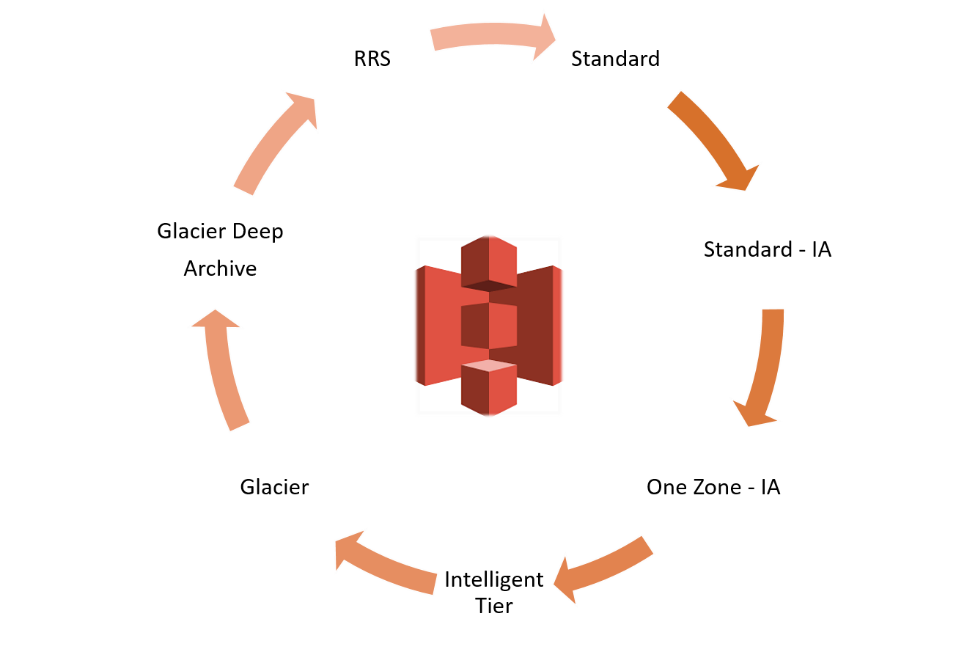
You can get buckets in one of the following three states:

* Unversioned (the default),
* Versioning-enabled,
* Versioning-suspended.

The versioning state applies to all (never some) of the objects in that bucket. The first time you enable a bucket for versioning, objects in it are thereafter always versioned and given a unique version ID.

* Objects stored in your bucket before you set the versioning state have a version ID of null.
* When you enable versioning, existing objects in your bucket do not change. What changes is how Amazon S3 handles the objects in future requests.
* The bucket owner (or any user with appropriate permissions) can suspend versioning to stop accruing object versions.
* When you suspend versioning, existing objects in your bucket do not change. What changes is how Amazon S3 handles objects in future requests.

### Lifecycle Management Basics



Lifecycle policies can be used to define actions that you want Amazon S3 to take during an object's lifetime such as:

* Transition objects to another storage class,
* Archive objects,
* Delete objects after a specified period of time.

A lifecycle policy can be used for either all objects or a subset of objects in the bucket by using a shared prefix.

A versioning-enabled bucket may have different versions of the same object, one current version, and zero or more previous versions. You can define actions specific to current and previous object versions by using a lifecycle policy as well.

### What is S3 Bucket Replication?

Replication is making a copy of the objects in the same AWS Region or different AWS Regions.

It allows automated, distributed copying of objects over buckets of Amazon S3. When enabled, each object uploaded to a specific S3 bucket will be replicated automatically to a specified destination bucket located in a given AWS Regions

Objects in S3 buckets can be replicated between different AWS Regions or within the same AWS Region. Here are the types of replication in S3:

* **Cross-Region Replication (CRR)** is used to copy objects in **different AWS Regions** across Amazon S3 buckets.
* **Same-Region Replication** **(SRR)** is used to copy objects in the **same AWS Region** across Amazon S3 buckets.
* Since replication requires versioning, a warning message would appear if the versioning is not enabled for the source bucket.

### Why Replication?

* **Replicate objects while retaining metadata** — You can use replication to **make copies of your objects that retain all metadata**, such as the original object creation time and version IDs. This capability is important if you need to ensure that your replica is identical to the source object.
* **Replicate objects into different storage classes** — You can use replication to **directly put objects into Glacier, DEEP ARCHIVE, or another storage** class in the destination bucket. You can also replicate your data to the same storage class and use lifecycle policies on the destination bucket to move your objects to a colder storage class as it ages.
* **Maintain object copies under different ownership** — Regardless of who owns the source object, you can tell Amazon S3 to **change replica ownership** to the AWS account that owns the destination bucket. This is referred to as the *owner's override* option. You can use this option to restrict access to object replicas.
* **Replicate objects within 15 minutes** — You can use S3 Replication Time Control (S3 RTC) to replicate your data in the same AWS Region or across different Regions in a predictable time frame. S3 RTC replicates 99.99 percent of new objects stored in Amazon S3 within **15 minutes** (backed by a service level agreement).

## S3 Static Website Hosting

Static Website Hosting is an exhibiting website that contains simple and static web components (e.g. HTML, CSS, images) without any server or database needs.

Static websites contain only static resources such as HTML, CSS, images, javascript, etc. They do not contain any application code and not have a database server. Because pages are stored in file storage, a static website provides exactly the same content on each request.

## If you have a static website with a few photos, a CSS or HTML file, you can use Amazon S3 to host the whole thing and get a URL to show the world. Amazon S3 will provide the origin for your website as well as storage for your static content.

## Static Website Hosting – Steps

## 

* **Create** an S3 **bucket**.
* Grant **Public Access** to the bucket while creating
* **Upload content** for hosting as a website.
* Don't forget to grant Public Access to the content of the website.
* Enable **Static website hosting** on S3 Bucket.
* You may do the same thing by adding the Policy and Access Control List after creating a bucket and uploading the items.  We 'll see also this method in in-class sessions/lab