



# S3 ESSENTIALS



# S3 = Simple Storage Service

## What is S3

### **Simplified Definition:**

An online, bulk storage service that you can access from almost any device.

### **AWS Definition:**

"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 user 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 users."

## **S3 Basics:**

### ***Components and Structure:***

#### **Basics:**

- (1) S3 = Simple Storage Service
- (2) It is AWS's primary storage service.
- (3) You can store any type of file in S3.



#### **Buckets:**

- (1) Root level "Folders" you create in S3 are referred to as ***buckets***.
- (2) Any "subfolder" you create in a bucket is referred to as a ***folder***.



S3 Folder

#### **Objects:**

- (1) Files stored in a bucket are referred to as ***objects***.



S3 Object

#### **Regions:**

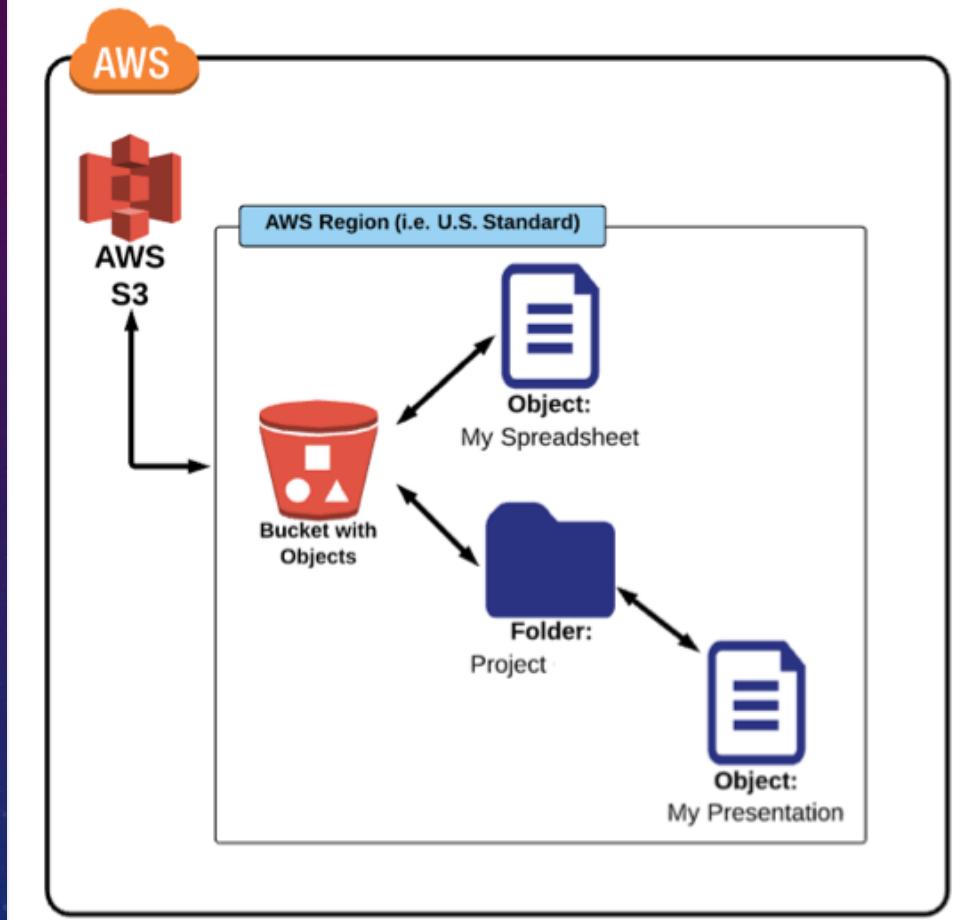
(1) When you create a bucket, you must select a specific region for it to exist. This means that ***any data you upload to the S3 bucket will be physically located in a data center in that region.***

(2) ***Best practice*** is to select the region that is physically ***closest to you***, to ***reduce transfer latency***.

#### **OR**

(3) If you are serving files to a ***customer*** based in a certain area of the world, ***create the bucket in a region closest to your customers*** (to reduce latency for your customers).

## Components & Structure:



## **Pricing/Cost Overview:**

**Free Tier** use is available for S3.

### **How are you charged for using S3?**

#### **(1) Storage Cost:**

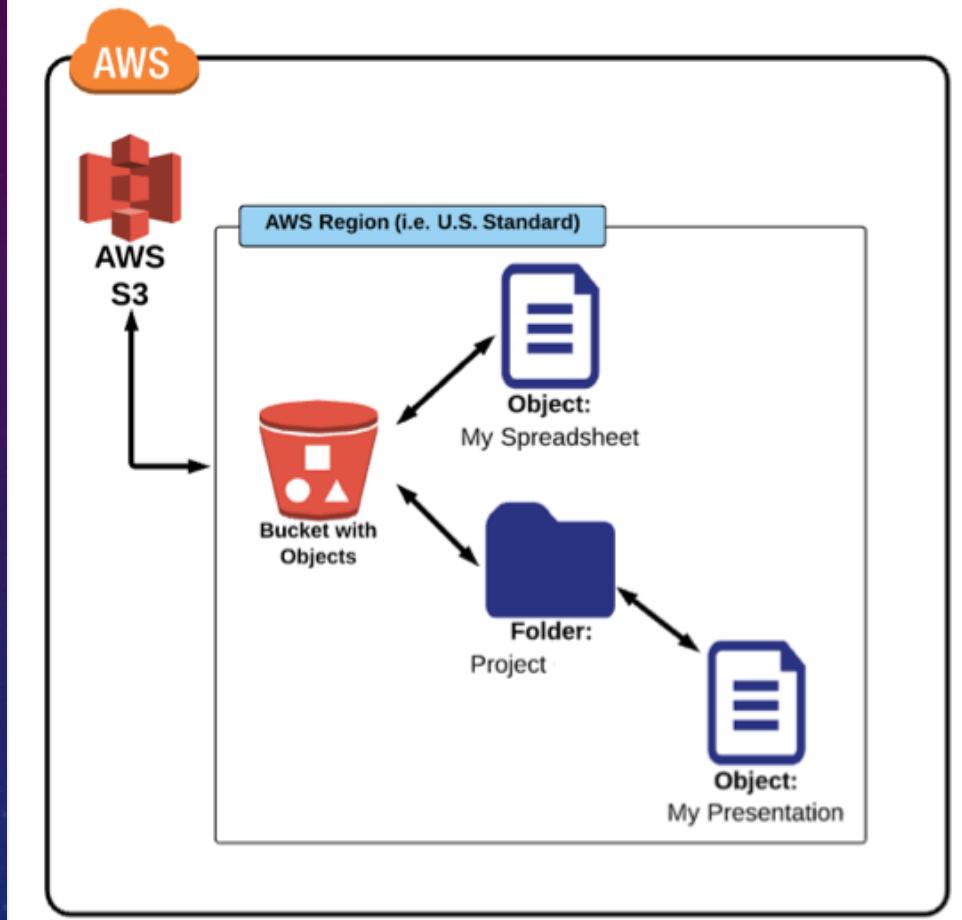
- Applies to data at rest in S3
- Charged per GB used
- Price per GB varies based on region and storage class

#### **(2) Request Pricing - moving data in/out of S3:**

- PUT
- COPY
- POST
- LIST
- GET
- Lifecycle Transitions Request
- Data Retrieval
- Data Archive
- Data Restore

***NOTE: Before doing any major usage of S3, you should make sure to review AWS's current pricing model to make sure you understand how much you will be required to pay.***

## Components & Structure:



## ***Creating an S3 Bucket:***

(1) Choose a bucket name:

### **Bucket names must follow a set of rules:**

- Bucket names must be unique across ALL of AWS.
- Bucket names must be 3 to 63 characters in length.
- Bucket names can only contain lowercase letters, numbers and hyphens.
- Bucket names must not be formatted as an IP address (e.g., 192.168.5.4).

(2) Select a region

**NOTE:** There are more “advanced” rules that allow for some varying formats, which can be found here:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html>

## ***Uploading (Import) an Object to a Bucket:***

- (1) Navigate into a bucket
- (2) Under “Actions” select “upload”
- (3) Select a file to upload
- (4) Click “Start Upload”

## ***Creating a Folder in a Bucket:***

- (1) Navigate into a bucket
- (2) Click on “Create Folder”
- (3) Give the folder a name

**NOTE:** Uploading an object directly into folder is the same process, just navigate into the folder first.

## **Buckets & Folders:**

### ***Bucket, Folder and Object Properties:***

#### **(1) Bucket Level Properties:**

- General Info
- Permissions
- Static Web Hosting
- Logging
- Events
- Versioning
- Lifecycle
- Cross-Region Replication
- Tags
- Requester Pays
- Transfer Acceleration

#### **(2) Folder Level Properties:**

- General Info
- Details

#### **(3) Object Level Properties:**

- General Info
- Details
- Permissions
- MetaData

# S3 Storage Classes

## What is a storage class?

(1) A **storage class** represents the "classification" assigned to each Object in S3.

**Available storage classes include:**

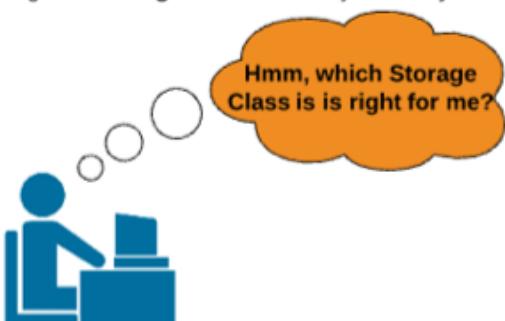
- Standard
- Reduced Redundancy Storage (RRS)
- Infrequent Access (S3-IA)
- Glacier

(2) Each **storage class** has varying attributes that dictate things like:

- Storage cost
- Object **availability**
- Object **durability**
- Frequency of access (to the object)

(3) Each object must be assigned a storage class ("standard" is the default class)

(4) You can change the **storage class** of an object at any time (*for the most part*).



### **Standard:**

- (1) Designed for general, all-purpose storage.
- (2) Is the default storage option.
- (3) **99.99999999% object durability** ("eleven nines").
- (4) **99.99% object availability**.
- (5) Is the most expensive storage class.

### **Reduced Redundancy Storage (RRS):**

- (1) Designed for non-critical, reproducible objects.
- (2) **99.99% object durability**.
- (3) **99.99% object availability**.
- (4) Is less expensive than the standard storage class.

### **Infrequent Access (S3-IA):**

- (1) Designed for objects that you do not access frequently, but must be immediately available when accessed.
- (3) **99.99999999% object durability**.
- (4) **99.90% object availability**.
- (5) Is less expensive than the standard/RRS storage classes.

### **Glacier:**

- (1) Designed for long-term archival storage.
- (2) May take several hours for objects stored in Glacier to be retrieved.
- (3) **99.99999999% object durability**
- (4) Is the cheapest S3 storage class (very low cost)

## ***Object Durability***

is the percent (%) over a one year time period  
that a file stored in S3 will ***NOT be lost.***

For object durability of 99.99999999% (**11 nines**) that means  
there is a 0.00000001% chance of a file in S3 being lost in a year.

**OR**

If you have 10,000 files stored in S3 (@ 11 nines durability),  
then you can expect to lose one file 10 million years.

## ***Object Availability***

Is the percent (%) over a one year time period  
that a file stored in S3 ***WILL be accessable.***

For object availability of 99.99% - that means there is a 0.01% chance  
that you won't be able to access a file stored in S3 in a year.

**OR**

For every 10,000 hours, you can expect a total of one hour for which a  
file may not be available to access.

## Setting/changing storage class:

- (1) By default, all new objects uploaded to S3 are set to the **Standard** storage class
- (2) If you want new objects to have a different storage class, then you need to set the proper settings prior to or during the upload process. You can do this by either:
  - Selecting another storage class during the upload process ("set details").
  - Using object **lifecycle policies** (covered in the next lesson).

- (3) For the following storage classes:

- Standard
- Reduced Redundancy Storage (RRS)
- Infrequent Access (S3-IA)

You can manually switch the objects storage class among them (at any time) by changing the storage class in the objects "properties".

- (3) To move an object to the **Glacier** storage class:

- You need to use object **lifecycles**.
- The change to Glacier may take 1-2 days to take effect.

### **Standard:**

- (1) Designed for general, all-purpose storage.
- (2) Is the default storage option.
- (3) **99.99999999% object durability** ("eleven nines").
- (4) **99.99% object availability.**
- (5) Is the most expensive storage class.

### **Reduced Redundancy Storage (RRS):**

- (1) Designed for non-critical, reproducible objects.
- (2) **99.99% object durability.**
- (3) **99.99% object availability.**
- (4) Is less expensive than the standard storage class.

### **Infrequent Access (S3-IA):**

- (1) Designed for objects that you do not access frequently but must be immediately available when accessed.
- (3) **99.99999999% object durability.**
- (4) **99.90% object availability.**
- (5) Is less expensive than the standard/RRS storage classes.

### **Glacier:**

- (1) Designed for long-term archival storage.
- (2) May take several hours for objects stored in Glacier to be retrieved.
- (3) **99.99999999% object durability**
- (4) Is the cheapest S3 storage class (very low cost)

# S3 Object Lifecycle

## What is an object lifecycle?

An object lifecycle is a **set of rules that automate** the migration of an object's storage class to a different storage class (or deletion), based on specified time intervals.

**For example:**

- (1) I have a work file that I am going to access every day for the next 30 days.
- (2) After 30 days, I may only need to access that file once a week for the 60 next days.
- (3) After which (90 days total) I will probably never access the file again but want to keep it just in case.

By using a lifecycle policy, I can **automate the process** of changing the files storage class to meet **my usage needs** AND keep my S3 storage **cost as low as possible**.

**Can you figure out what the lifecycle policy would look like?  
What storage classes should be used for each time interval?**

# Object Lifecycle:

## **The scenario:**

- (1) I have a work file that I am going to access every day for the next 30 days.
- (2) After 30 days, I may only need to access that file once a week for the 60 next days.
- (3) After which (90 days total) I will probably never access the file again but want to keep it just in case.

***What is the best solution to meet the usage needs and minimize storage cost?***

## **The Solution:**

### **Day 0-29 (30 days):**

- (1) Usage needs = Very frequent
- (2) "Best fit" Storage class = Standard
- (3) Cost = highest cost tier



### **Day 30-89 (60 days):**

- (1) Usage needs = Infrequently
- (2) "Best fit" Storage class = Infrequent Access
- (3) Cost = middle cost tier

### **Day 90+:**

- (1) Usage needs = Most likely never needed
- (2) "Best fit" Storage class = Glacier
- (3) Cost = lowest cost tier

## **Lifecycle Management:**

(1) Lifecycles functionality is located on the bucket level.

(2) However, a lifecycle policy can be applied to:

- The entire bucket (applied all the objects in the bucket)
- One specific folder within a bucket (applied to all the objects in that folder).
- One specific object within a bucket

(3) You can always delete a lifecycle policy or manually change the storage class back to whatever you like.

### ***Let's create the lifecycle!***

#### **Day 0-29 (30 days):**

- (1) Usage needs = Very frequent
- (2) "Best fit" Storage class = Standard
- (3) Cost = highest cost tier

#### **Day 30-89 (60 days):**

- (1) Usage needs = Infrequently
- (2) "Best fit" Storage class = Infrequent access
- (3) Cost = middle cost tier

#### **Day 90+:**

- (1) Usage needs = Most likely never needed
- (2) "Best fit" Storage class = Glacier
- (3) Cost = lowest cost tier

# S3 Permissions

## What are S3 Permissions?

S3 permissions are what allow you to have ***granular control*** over who can view, access and use specific buckets and objects.

(1) Permissions functionality can be found on the bucket ***AND*** object level.

(2) On the bucket level you can control (for each bucket individually):

- List:** Who can see the bucket name.
- Upload/Delete:** Objects to (upload) or in the bucket (delete).
- View Permissions**
- Edit Permissions:** Add/edit/delete permissions

***NOTE: Bucket level permission are generally used for "internal" access control***

(3) On the Object level, you can control: (for each object individually)

- Open/Download**
- View Permissions**
- Edit Permissions**

***NOTE: You can share specific objects (via a link) with the anyone in the world.***

## ***Sharing an S3 object with the world:***

(1) On the object, create the following permission:

- Grantee = Everyone
- "Check" Open/Download

(2) Under "Actions", select "Make Public"

(3) The "Link" under ***Properties*** is now live and anyone that has it can directly download the object.

***NOTE: To remove public access to the object, either delete the permission***

# S3 Versioning

## What is S3 Versioning?

S3 versioning is a feature that keeps track of and stores all old/new versions of an object so that you can access and use an older version if you like.

- (1) Versioning is either ON or OFF.
- (2) Once it is turned ON, you can only "suspend" versioning. It cannot be fully turned OFF.
- (3) Suspending versioning only prevents versioning going forward. All previous objects with versions will still maintain their older versions.
- (4) Versioning can only be set on the bucket level and applies to ALL objects in the bucket.