AWS S3

A SECURE, HIGHLY-AVAILABLE
& DURABLE STORAGE SERVICE
IN THE CLOUD

AWSFUNDAMENTALS.COM

INTRODUCTION 👋

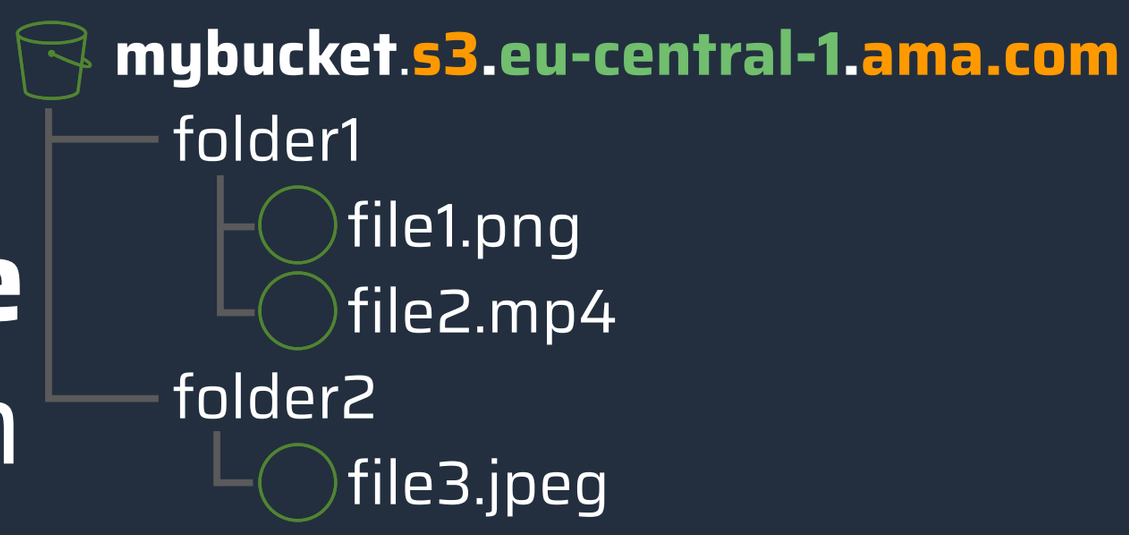
Amazon’s **S**imple **S**torage **S**ervice (S3) is a storage service for any amount of data and different use cases, including:

- static websites
- backups & archives
- data lakes
- IoT devices
- big data analytics

With S3’s **management features** you’re enabled to optimise the storage of and access to your data easily and in a fine-grained way.

BUCKETS

Buckets are the most fundamental part of S3.
Think about it like a folder on your computer - but with **unlimited storage** and **number of files**.



A bucket needs a **unique naming** - not only within your account but globally.

Buckets can be configured in different ways, for example with dedicated access policies or encryption modes.
Additionally, each file in a **bucket** is considered an **object** and can have its own **metadata** and **configuration**.

STORAGE CLASSES

Amazon offers different types of storages, which help you to find a good trade-off between **availability and durability based on your access patterns** and **costs**.

The types can be split into **three major groups**.

Additionally, there's the **Intelligent Tiering** storage class for data with unknown, changing or unpredictable access patterns.

AWSFUNDAMENTALS.COM

STORAGE MANAGEMENT ⚙️

S3 not only enables you to make use of different storage classes to reduce costs, but also offers additional management features like:

- **Lifecycle Policies**
- **Object Locks**
- **Replication**
- **Batch Operations**

Those features can help you for example to meet requirements due to regulations or compliance.

AWSFUNDAMENTALS.COM

LIFECYCLE POLICIES 🌱

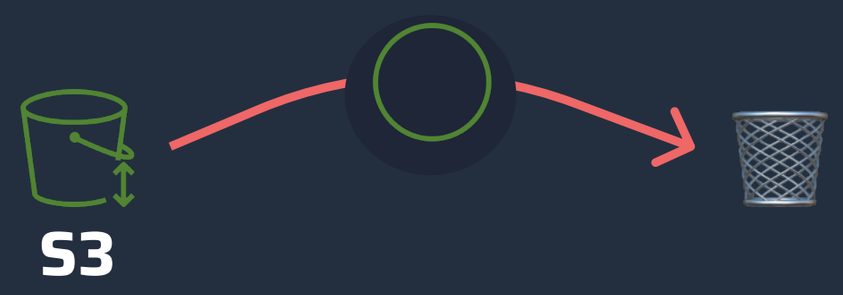
Objects stored within your buckets sometimes do have changing requirements within their lifecycle. With Lifecycle Policies, you can automatically adapt to those requirements.

A lifecycle policy can contain multiple rules, which in turn specify **actions** that **apply to a group of objects** when a certain condition is met.

There are two types of action:

- **Transition** - moving object to another storage class
- **Expiration** - deleting objects

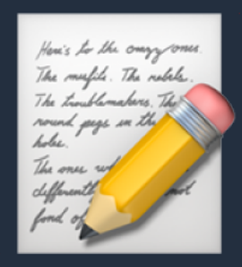
move to Glacier 30 days after creation



delete 90 days after creation

AWSFUNDAMENTALS.COM

BUCKET POLICIES



Bucket policy enable you to grant access permissions to your bucket and to the objects inside of it.
Policies are JSON-based and either **allow** or **deny** certain **actions** on a **Resource** for a given **principal**.

In the example, we...

- grant a **CloudFront Origin Access Identity Principal**
- **read permissions**
- for **all objects** inside our **target bucket**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudFrontAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront [ ... ]"
      },
      "Action": [
        "s3:List*",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::mybucket/*",
        "arn:aws:s3:::mybucket"
      ]
    }
  ]
}
```

AWSFUNDAMENTALS.COM

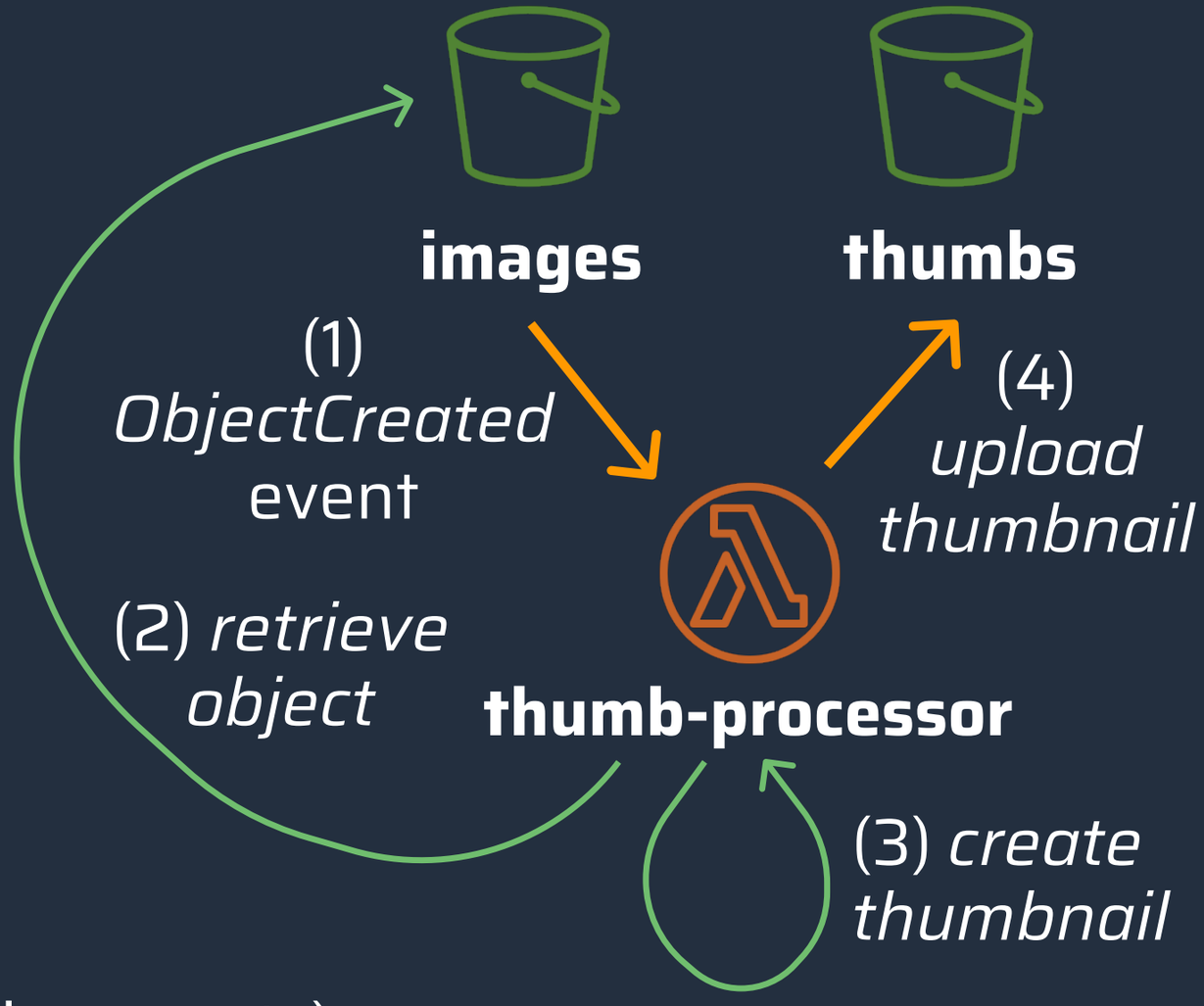
EVENT NOTIFICATIONS 📥

There are a lot of use cases that require triggering processes in the case of events happening at S3.

Prominent example: creating a thumbnail after an image was uploaded to a bucket.

S3 can send notification messages to different destinations like

- SNS Topics
- SQS Queues
- Lambda Functions



and for different events, including:

- **creation** or **celetion** of objects
- **lifecycle executions** (e.g. expiration events)
- object **taggings**
- **ACL PUT**s

With this, we can easily cover our example:
ObjectCreated events for objects with an **image suffix** at a specific bucket will **trigger a Lambda function**, which in turn will create a thumbnail for the uploaded image and upload it to another bucket.

BATCH OPERATIONS ⚡

With single API requests, you are able to manage billions of objects.

A batch job will execute a specified operation on every object that’s included in the job description. Running jobs can be monitored programatically or via the AWS console.

ACCESS DENIED 

When working with AWS, especially in the beginning, you **will** face this message at least from time to time.

Mostly, AWS API error responses will excactly point you the missing permission so you can easily extend your policy.

Sometimes it's not that easy and you'll need to get back to the documentation to find out about required permissions. A great resource is the **Actions, resources, and condition keys for AWS services** with can be found at the **Service Authorization Reference** and does lists every IAM action, resource and conditions for every AWS service.

What you should never do: just blindly extending your policy with all permissions for your target AWS service by adding wildcards for actions and resources, e.g. **action: ["dynamodb:*"]** and **resource: ["*"]**.

You won’t gain any learnings for IAM and you’re distributing unnecessary permissions which can lead to critical incidents.

AWSFUNDAMENTALS.COM

OBJECT LOCKS 

S3 also helps you to keep your files as safe as possible with Object Locks.

Mnemonic: **WORM**

It allows you to store files using a **W**rite-**O**nce-**R**ead-**M**any model.

Easily ensure that for example...
- backups
- log files

can't be deleted or manipulated

You can choose between two different retention modes:

- **Compliance** - no deletes or overwrites possible for the dureation of the retention period
- **Goverance** - overwrites/deletes are only possible with specific rights

REPLICATION

With replication, you can asynchronously copy objects across S3 buckets. You're not bound to buckets that are owned by your account.

Replication helps you to easily create **identical copies** (including the **metadata**), e.g. to create redundancy for backups.

You can even...

- replicate the copy into a **different storage class**
- move them to a **different ownership**
- or store them over **multiple AWS regions**

BLOCK PUBLIC ACCESS ❌

By default, new buckets and objects are not publicly accessible through the internet.

As users can modify bucket policies and object permissions, public access can be enabled.

With **Block Public Access**, account and bucket owners can easily setup centralized controls that are **enforced regardless of how the resources are created**.

AWSFUNDAMENTALS.COM

VERSIONING 👥

For a lot of use cases, it's necessary that files are not simply overwritten but only created with a new version, so that you keep older versions & can easily rollback to a previous state.

Example: your Terraform's infrastructure state files.

S3 also got you covered here: you can enable versioning per bucket with a single option.

⭐ **Bonus**: Versioning can be combined with Lifecycle Policies.

AWSFUNDAMENTALS.COM

FREE TIER 

AWS' free tier for S3 includes:

- **5 GB** of standard storage
- **20.000 GET** requests
- **2.000 PUT** requests

💡 You'll be notified via e-mail when you're approaching the limits of the free tier.

PRICING 💰

You’re paying for **storing objects**, making **requests against your buckets**, **data transfer** & advanced features.

- for storage & requests, the rates **can highly vary** based on the storage class.
- traffic charges are based on the source & destinations, e.g. from S3 to the Internet you’ll pay $0.09 per GB but transfer from the Internet to S3 is free, as well as outgoing traffic from S3 to CloudFront.

As with other services, rates can vary also between regions.

Interested in Learning AWS for the real world?

Check out our **newsletter**, **fundamentals book**, and **socials** 🪄

tpschmidt

@tpschmidt_

@sandro_vol

alessandro-volpicella

AWSFUNDAMENTALS.COM