 Amazon S3

## What:

Scalable object storage service for storing and retrieving any amount of data.

## How:

Upload, retrieve, and manage data using the console, CLI, or API.

## How Much:

Pay based on storage usage, data transfer, and number of requests.

## Objective:

Create an S3 bucket, upload files, experiment with public vs private access, enable static website hosting, and apply common production best practices (CORS, lifecycle, versioning, encryption).

Estimated time: 20–45 minutes

Prerequisites: AWS account, Console access, (optional) AWS CLI configured (aws configure) and basic CLI familiarity.

# Step 1 — Create the bucket (Console + CLI)

**Console:**  
1. Open S3 in the AWS Console → Create bucket.  
2. Bucket name: my-unique-bucket-name-12345.  
3. Region: pick us-west-2 (or your preferred region).  
4. Leave Block public access ON for now (we’ll change later if you intentionally make it public).  
5. Click Create bucket.  
  
**CLI:**  
*aws s3api create-bucket --bucket my-unique-bucket-name-12345 --region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2*

# Step 2 — Upload files (Console + CLI)

**Console:**

Open the bucket → Upload → drag & drop index.html, error.html, images, etc. → Click Upload.

**CLI:**  
*aws s3 cp index.html s3://my-unique-bucket-name-12345/index.html  
aws s3 sync ./site s3://my-unique-bucket-name-12345/*

# Step 3 — Public vs Private: Blocking public access & bucket policy

By default, S3 Block Public Access is enabled. To make objects public, disable block public access and apply a bucket policy.  
  
Example bucket policy:  
***{  
 "Version": "2012-10-17",  
 "Statement": [  
 {  
 "Effect": "Allow",  
 "Principal": "\*",  
 "Action": "s3:GetObject",  
 "Resource": "arn:aws:s3:::my-unique-bucket-name-12345/\*"  
 }  
 ]  
}***

# Step 4 — Enable Static Website Hosting

**Console:**

Bucket → Properties → Static website hosting → Enable → Index document = index.html, Error document = error.html.  
  
**CLI:**  
aws s3 website s3://my-unique-bucket-name-12345 --index-document index.html --error-document error.html

# Step 5 — Serve via CloudFront (recommended)

Steps:  
1. Create CloudFront distribution → Origin = your S3 bucket.  
2. Restrict bucket access to CloudFront.  
3. Add ACM certificate for HTTPS.  
4. Point Route 53 domain to CloudFront.  
*Why*: CloudFront gives HTTPS, caching, DDoS protection, and keeps S3 private.

# Step 6 — CORS configuration

1.Click on the bucket and Choose **Permissions.**

2. In the **Cross-origin resource sharing (CORS)** section, choose **Edit**.

3.In the **CORS configuration editor** text box, type or copy and paste a new CORS configuration, or edit an existing configuration.

Example XML:

<CORSConfiguration>  
 <CORSRule>  
 <AllowedOrigin>https://example.com</AllowedOrigin>  
 <AllowedMethod>GET</AllowedMethod>  
 <AllowedMethod>HEAD</AllowedMethod>  
 <AllowedHeader>\*</AllowedHeader>  
 </CORSRule>  
</CORSConfiguration>

# Step 7 — Versioning & Lifecycle

Enable versioning:  
*aws s3api put-bucket-versioning –bucket my-unique-bucket-name-12345 –versioning-configuration Status=Enabled*  
Lifecycle rule example: Transition to STANDARD\_IA after 30 days, expire after 365 days.

# Step 8 — Test & validate

Open the static site endpoint in browser or use curl:  
*curl -I http://my-unique-bucket-name-12345.s3-website-us-west-2.amazonaws.com*

# Step 9 — Troubleshooting

- 403 AccessDenied → Check Block Public Access and bucket policy.  
- 404 Not Found → Ensure index.html exists.  
- CORS issues → Fix AllowedOrigin in CORS config.  
- HTTPS required → Use CloudFront + ACM.

# Step 10 — Cleanup

**Console:**

1. Click on the bucket and choose empty buckets.

2. On the **Empty bucket** page, confirm that you want to empty the bucket by entering the bucket name into the text field, and then choose **Empty**.

3. In the buckets list, select the option button next to the name of the bucket that you want to delete, and then choose **Delete** at the top of the page.

4.On the **Delete bucket** page, confirm that you want to delete the bucket by entering the bucket name in the text field, and then choose **Delete bucket**.

**CLI:**

Delete objects:  
*aws s3 rm s3://my-unique-bucket-name-12345 --recursive*  
  
Delete bucket:  
*aws s3api delete-bucket --bucket my-unique-bucket-name-12345 --region us-west-2*

Note: If versioning was enabled, you must delete all versions first.

# Best Practices & Quick Tips ✅

* Prefer **CloudFront + Origin Access Control** over making buckets public.
* Use **least privilege** for any IAM principals that access the bucket.
* Enable **bucket encryption**, **versioning**, and **lifecycle rules** for data governance.
* Keep logs (access + CloudTrail) for audits.
* Use **signed URLs** for temporary private access to objects.
* Use naming conventions and tagging for cost/allocation tracking.



