This is an excellent end-to-end project that touches on core S3 features: **Hosting, Security, Versioning, Cost Management, and Auditing**.

Here is the complete, step-by-step guide. You will need two separate buckets for this project: one for the **website** and one for the **logs**.

🏗️ S3 Project: Full Deployment and Management

Prerequisites

1. **AWS Account:** Active AWS account with IAM User credentials.
2. **Files:**
   - One generic file to upload (e.g., report.pdf).
   - Two basic website files: index.html and error.html (the content doesn't matter, just the names).

Phase 1: Creation, Versioning, and Upload (The Foundation)

Step 1: Create the Primary (Website) Bucket (e.g., my-project-website-2025)

1. Navigate to the **S3 Console**.
2. Click **Create bucket**.
3. **Bucket Name:** Choose a globally unique name (must be all lowercase, e.g., my-project-website-2025).
4. **AWS Region:** Select your preferred region.
5. **Block Public Access:** Leave all options **checked** (Blocked) for now. We will unblock this later.

6. Click **Create bucket**.

Step 2: Enable Versioning (The Safety Net)

1. Open your primary bucket.
2. Go to the **Properties** tab.
3. Scroll to **Bucket Versioning** and click **Edit**.
4. Select **Enable Versioning**.
5. Click **Save changes**.

> 💡 **Why Versioning is Important:** This protects you against accidental **deletion** or **overwriting** of your files, as S3 will keep every old version, ensuring recovery is possible.

Step 3: Initial Uploads and Testing Versioning

1. Click the **Objects** tab and click **Upload**.
2. Upload your first file, report.pdf.
3. Now, upload a **different version** of the same file, *also named* report.pdf.
   - Go to the **Objects** tab, select the file, and look under the **Versions** toggle (**Show versions**). You will see both copies listed with unique **Version IDs**.

Phase 2: Temporary Sharing and Cost Management

Step 4: Share a File Temporarily (Pre-signed URL)

1. In the **Objects** tab, select one of the uploaded files (e.g., report.pdf).
2. Click the **Actions** dropdown.
3. Select **Share with a pre-signed URL**.
4. Set the **Expiration** (e.g., **60 minutes**).
5. Click **Generate pre-signed URL**.
6. **Copy the URL.** Anyone with this link can view the file for 60 minutes, even though the

bucket is still private.

Step 5: Configure Lifecycle Management (Cost Control)

This rule moves data to cheaper storage and then deletes old data to save costs.

1. Go to the bucket's **Management** tab.
2. Click **Create lifecycle rule**.
3. **Rule name:** Archive-and-Delete-Old-Versions.
4. **Rule scope:** Choose **Apply to all objects in the bucket**.
5. **Lifecycle rule actions:**
   - **Transition Action:** Check **Transition current versions of objects between storage classes**.
     - **Days after object creation:** Enter **30**.
     - **Choose storage class:** Select **Standard-IA** (Infrequent Access).
   - **Expiration Action (Current Versions):** Check **Permanently delete current versions of objects**.
     - **Days after object creation:** Enter **365**.
   - **Expiration Action (Previous Versions):** Check **Permanently delete previous versions of objects**.
     - **Days after object creation:** Enter **90**.
6. Click **Create rule**.

Phase 3: Static Website Hosting and Logging (Public Access)

Step 6: Create the Target (Log) Bucket (e.g., my-project-website-logs-2025)

Logs **MUST** go to a separate bucket in the same region to avoid an infinite logging loop.

1. Click **Create bucket** again.
2. **Bucket Name:** (e.g., my-project-website-logs-2025).
3. **Block Public Access: KEEP ALL OPTIONS CHECKED.** This log bucket must remain

private.
4. Click **Create bucket**.

Step 7: Configure Log Bucket Policy (Grant Log Delivery Permission)

This policy allows the S3 logging service to write logs into your new private log bucket.

1. Go to your **Log Bucket** (my-project-website-logs-2025).
2. Go to **Permissions** $\rightarrow$ **Bucket Policy** $\rightarrow$ **Edit**.
3. Paste the following policy (replace YOUR-LOG-BUCKET-NAME):
   JSON
   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "S3ServerAccessLogsPolicy",
         "Effect": "Allow",
         "Principal": {
           "Service": "logging.s3.amazonaws.com"
         },
         "Action": "s3:PutObject",
         "Resource": "arn:aws:s3:::YOUR-LOG-BUCKET-NAME/*"
       }
     ]
   }
   ```
4. Click **Save changes**.

Step 8: Enable Public Access on Website Bucket

1. Go back to your **Website Bucket** (my-project-website-2025).
2. Go to **Permissions** $\rightarrow$ **Block public access (bucket settings)** $\rightarrow$ **Edit**.
3. **Uncheck** the option **Block all public access** and confirm. **Save changes.**

Step 9: Set Public Read Bucket Policy on Website Bucket

This policy grants public read access, allowing the world to see your website files.

1. In the **Website Bucket** $\rightarrow$ **Permissions** $\rightarrow$ **Bucket Policy** $\rightarrow$ **Edit**.
2. Paste the following policy (replace YOUR-WEBSITE-BUCKET-NAME):
   JSON
   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "PublicReadGetObject",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::YOUR-WEBSITE-BUCKET-NAME/*"
       }
     ]
   }
   ```
3. Click **Save changes**.

Step 10: Enable Static Website Hosting

1. Go to the **Website Bucket** $\rightarrow$ **Properties** tab.
2. Scroll to **Static website hosting** and click **Edit**.
3. Select **Enable**.
4. **Index document:** Enter index.html.
5. **Error document:** Enter error.html.
6. Click **Save changes**.
7. Copy the **Bucket website endpoint** URL.

Step 11: Enable Server Access Logging (Auditing)

1. In the **Website Bucket** $\rightarrow$ **Properties** tab.

2. Scroll to **Server access logging** and click **Edit**.
3. Select **Enable logging**.
4. **Target bucket:** Choose your **Log Bucket** (my-project-website-logs-2025).
5. Click **Save changes**.

🚨 Important Takeaways and Clean-up

Topic	**Important Detail to Take Care Of**
Public Access	**Never** disable Block Public Access on a bucket unless you *absolutely* intend to host a public website. This is the **number one cause of S3 data breaches.**
Logging	The target log bucket **must** be a different bucket, in the **same region**, and must have the correct **Bucket Policy** to allow logging.s3.amazonaws.com to write objects.
Versioning & Cost	Enabling **Versioning** protects data, but it can increase costs by storing every file version. **Lifecycle Management** is essential to delete older, non-current versions to mitigate this cost increase.
HTTPS	The S3 website endpoint is **HTTP only**. For secure HTTPS access (which is mandatory for modern sites), you must configure **Amazon CloudFront** in front of your S3 bucket.
Clean Up	After the project, **delete both buckets** (the website bucket and the log bucket) to stop all billing for storage and requests.