

Web-Hosting Using AWS S3 Bucket

Project Description: The goal of this project is to host a simple static website using AWS S3. The website consists of basic HTML and CSS files (e.g., index.html, style.css). Using AWS S3, the files are uploaded to a bucket and configured to be publicly accessible, enabling the site to be viewed in any web browser. This project demonstrates the use of cloud storage, bucket permissions, and static website hosting on AWS.

Skills Learned

- AWS S3 Bucket Creation and Management
- Configuring Bucket Policies and Permissions for public access
- Static Website Hosting on AWS S3
- Using Linux/Windows CLI to upload and manage files

Tools and Technologies Used

- AWS S3 — for storage and hosting
- AWS CLI — for command-line management of S3 buckets
- HTML/CSS — for the static website
- Linux/Windows Terminal — for executing AWS CLI commands

Prerequisites

Linux machine with internet access.

AWS CLI installed and configured (aws configure).

IAM user with s3:fullpermission permission for the target bucket.

Step 1: Create an S3 Bucket After logging into the AWS console, navigate to the S3 service and click "Create bucket." Provide a globally unique name for your bucket.

Create bucket [info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [info](#)
project01-bucket01

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [info](#)
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Set default and create bucket

Default encryption [info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [info](#)

☒ **Server-side encryption with Amazon S3 managed keys (SSE-S3)**
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page](#).

☐ **Server-side encryption with AWS Key Management Service keys (SSE-KMS)**

☐ **Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)**

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ **Disable**

☒ **Enable**

Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

Allow S3 Bucket Publick access.

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Vector buckets

Access Grants

Access Points (General Purpose Buckets, Fsx file systems)

Access Points (Directory Buckets)

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Edit Block public access (bucket settings) [info](#)

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☒ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

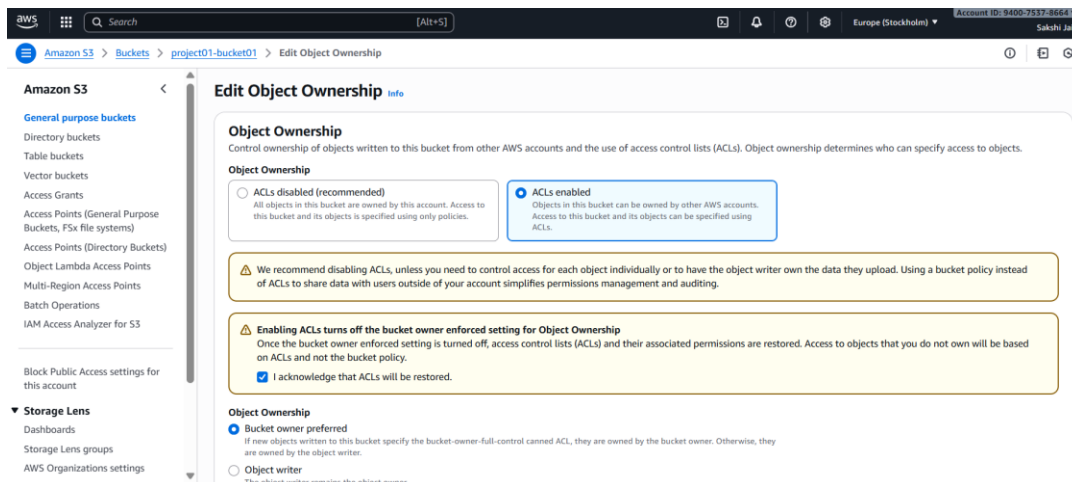
☒ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.

☒ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

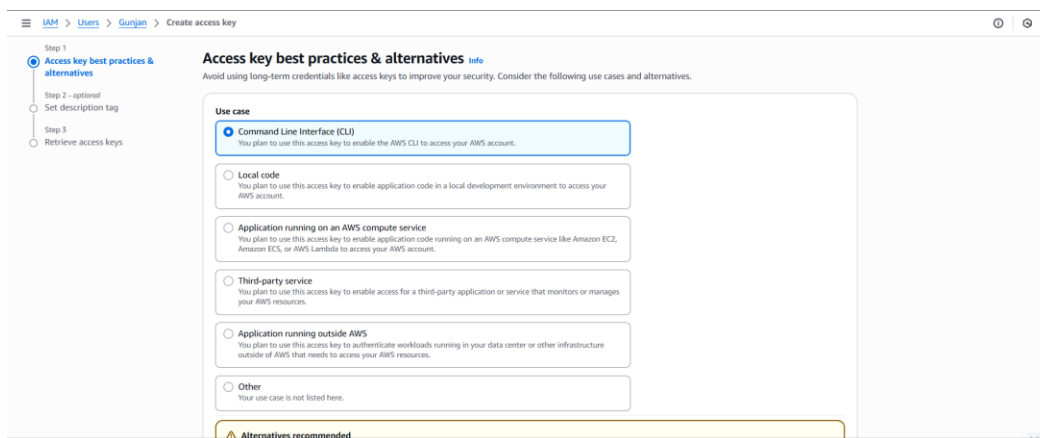
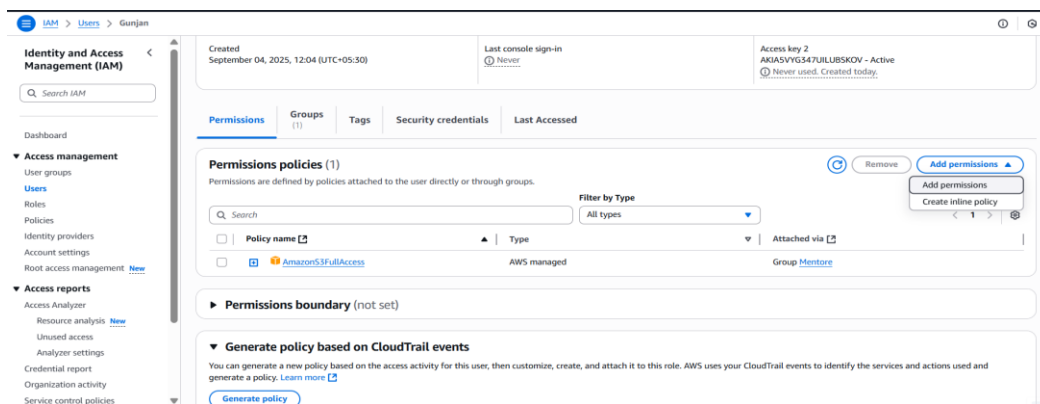
☒ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

[Cancel](#) [Save changes](#)

Set Object Ownership so that Object inside bucket would be public



Step 2: Now create IAM user and give S3 Full access and generate Access key for that user.



Step 3: Configure AWS to Windows CLI

User your user access key here

```
C:\Users\sjain403>aws configure
AWS Access Key ID [*****CYUH]: AKIA5VYG347UILUBSKOV
AWS Secret Access Key [*****odJh]: xVylsW7QIzD9+nduTOHswvOUdwCx6+l+e3W3huD
Default region name [None]:
Default output format [None]:

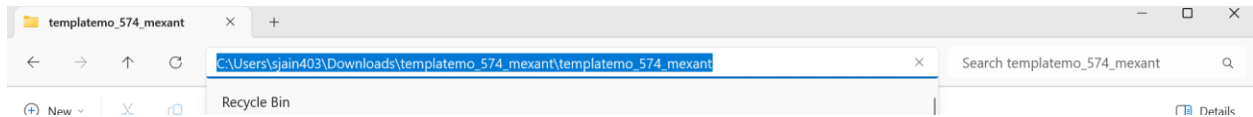
C:\Users\sjain403>
```

Here you can see the list of s3 buckets

```
C:\Users\sjain403>aws s3 ls
2025-09-29 22:51:03 mentore-solution001
2025-10-13 11:11:05 project01-bucket01

C:\Users\sjain403>aws --version
aws-cli/2.27.7 Python/3.13.2 Windows/11 exe/AMD64
```

Step 4: Download any static website from this <https://templatemo.com/> and Unzip it. Then Copy the path of Unzip folder



Now Goto command prompt and use command

aws s3 sync “your template folder path” s3://your-bucket-name

Upload Your Website Files via Windows CLI

```
C:\Users\sjain403>aws s3 sync "C:\Users\sjain403\Downloads\templatemo_574_mexant\templatemo_574_mexant" s3://project01-bucket01
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\css\animate.css to s3://project01-bucket01/assets/css/animate.cs
s
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\cta-bg.jpg to s3://project01-bucket01/assets/images/cta-b
g.jpg
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\about-us.html to s3://project01-bucket01/about-us.html
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\heading-bg.jpg to s3://project01-bucket01/assets/images/h
eading-bg.jpg
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\loading.gif to s3://project01-bucket01/assets/images/load
ing.gif
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\client-01.png to s3://project01-bucket01/assets/images/cl
ient-01.png
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\close.png to s3://project01-bucket01/assets/images/close.
png
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\css\owl.css to s3://project01-bucket01/assets/css/owl.css
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\css\flex-slider.css to s3://project01-bucket01/assets/css/flex-s
lider.css
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\css\templatemo-574-mexant.css to s3://project01-bucket01/assets/
css/templatemo-574-mexant.css
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\logo.png to s3://project01-bucket01/assets/images/logo.pn
g
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\next.png to s3://project01-bucket01/assets/images/next.pn
g
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\prev.png to s3://project01-bucket01/assets/images/prev.pn
g
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\css\fontawesome.css to s3://project01-bucket01/assets/css/fontaw
esome.css
upload: Downloads\templatemo_574_mexant\templatemo_574_mexant\assets\images\service-details-01.jpg to s3://project01-bucket01/assets/
```

Step 5: Goto s3 and check you will upload data here.

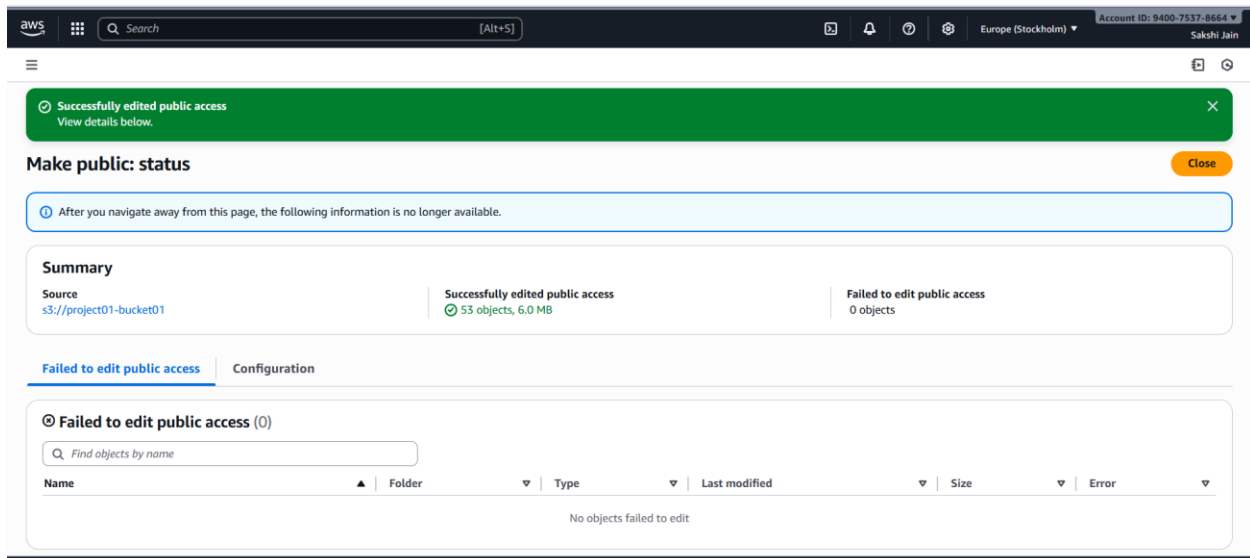
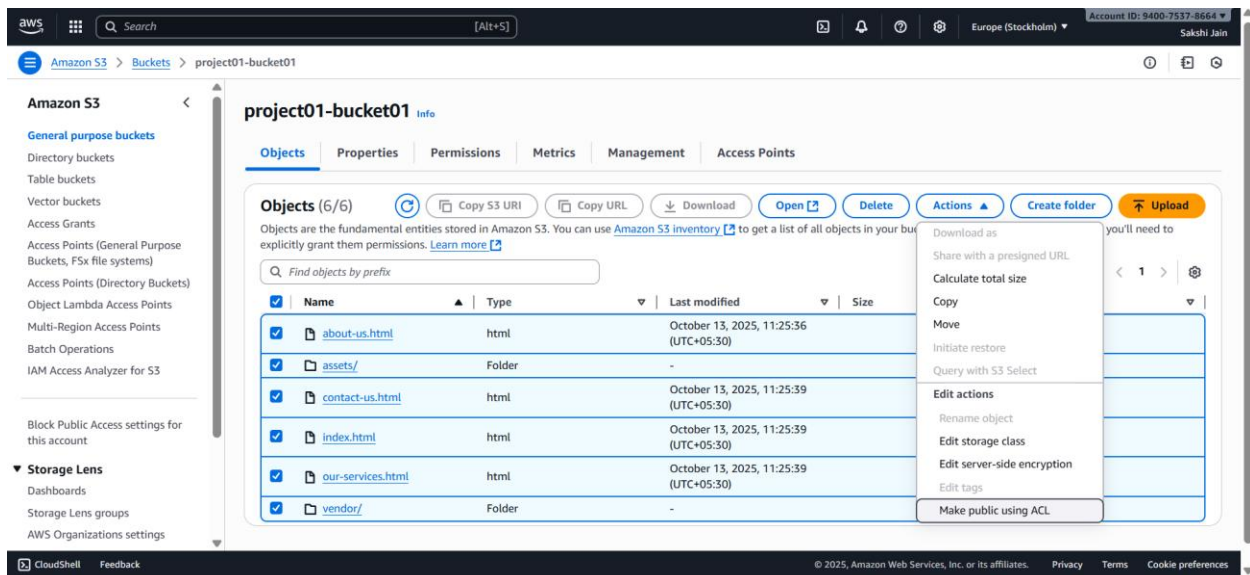
The screenshot shows the Amazon S3 console interface. On the left, the 'Amazon S3' sidebar is visible with various navigation options. The main content area displays the 'project01-bucket01' bucket. The 'Objects' tab is selected, showing a list of 6 objects. The objects are:

Name	Type	Last modified	Size	Storage class
about-us.html	html	October 13, 2025, 11:25:36 (UTC+05:30)	11.8 KB	Standard
assets/	Folder	-	-	-
contact-us.html	html	October 13, 2025, 11:25:39 (UTC+05:30)	10.0 KB	Standard
index.html	html	October 13, 2025, 11:25:39 (UTC+05:30)	23.9 KB	Standard
our-services.html	html	October 13, 2025, 11:25:39 (UTC+05:30)	13.7 KB	Standard
vendor/	Folder	-	-	-

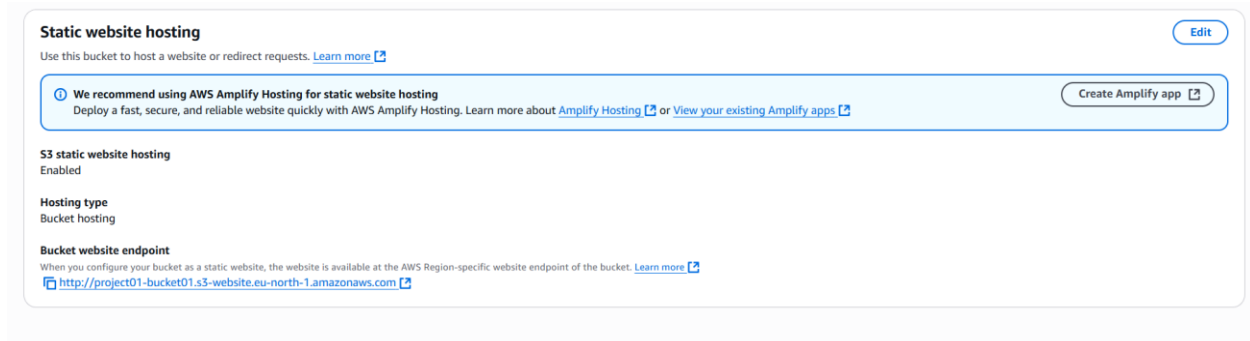
Now Goto properties and enable web hosting

The screenshot shows the 'Edit static website hosting' page for the 'project01-bucket01' bucket. The 'Static website hosting' section is expanded, and the 'Enable' radio button is selected. The 'Hosting type' section is also expanded, and the 'Host a static website' radio button is selected. A blue information box states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access.' The 'Index document' field is set to 'index.html' and the 'Error document - optional' field is set to 'error.html'.

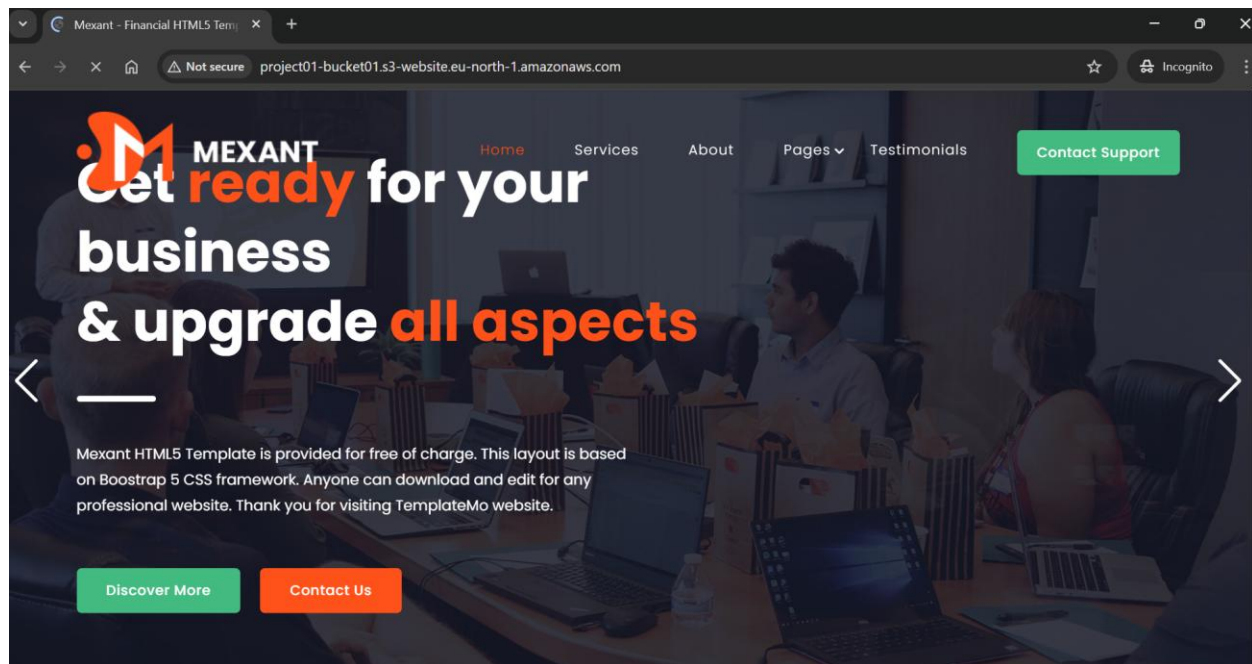
Make all object public using CLI



Step 6: Copy the website link <http://project01-bucket01.s3-website.eu-north-1.amazonaws.com>



Open the link you can see your hosted website



Project Summary

The Automated Backup of Linux Files to AWS S3 project focuses on ensuring the safety and reliability of critical Linux system data by automating the backup process to cloud storage. The project involves creating a Bash shell script that compresses important directories or files into a timestamped archive and uploads them to a specified AWS S3 bucket using the AWS CLI. To make the process fully automated, the script can be scheduled using cron jobs to run at regular intervals, such as daily or weekly.

Conclusion

The Automated Backup of Linux Files to AWS S3 project shows how important Linux files can be safely backed up to the cloud automatically. Using Bash scripts, AWS CLI, and cron, the project compresses files and uploads them to an S3 bucket with timestamps.

Name:

Sakshi Jain

Umesh Chimankar

Jay Soni

Nikita Binnar

Vaishavi Kumbhar

Kaveri Kanawade