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

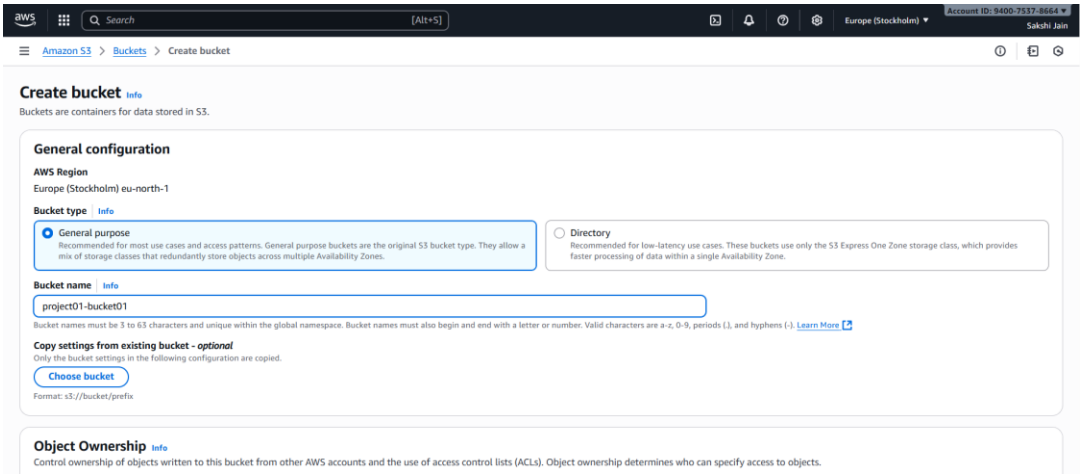
Prerequisites

Linux machine with internet access.

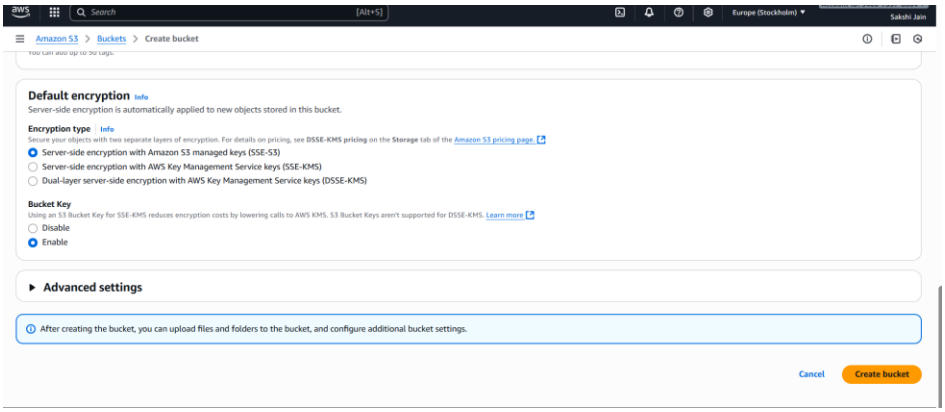
AWS CLI installed and configured (aws configure).

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

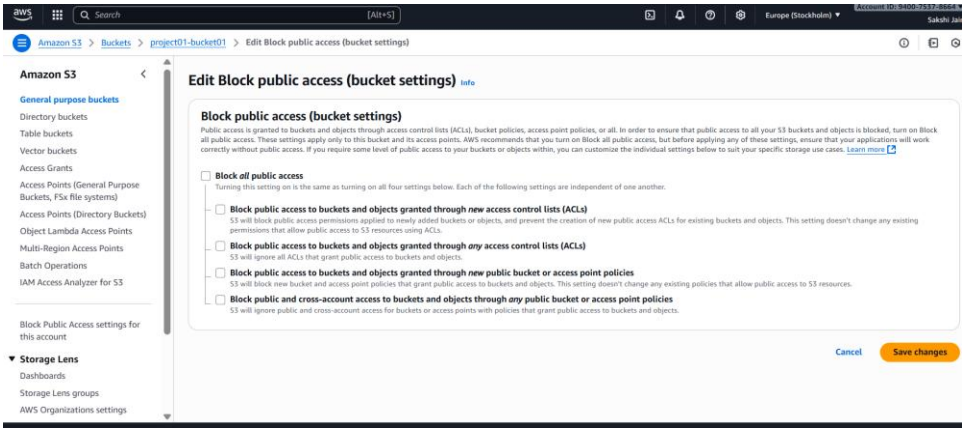
Step 1: Create an S3 Bucket After logging into the AWS console, navigate to the S3 service and click "Create bucket." Provide a globally unique name for your bucket.



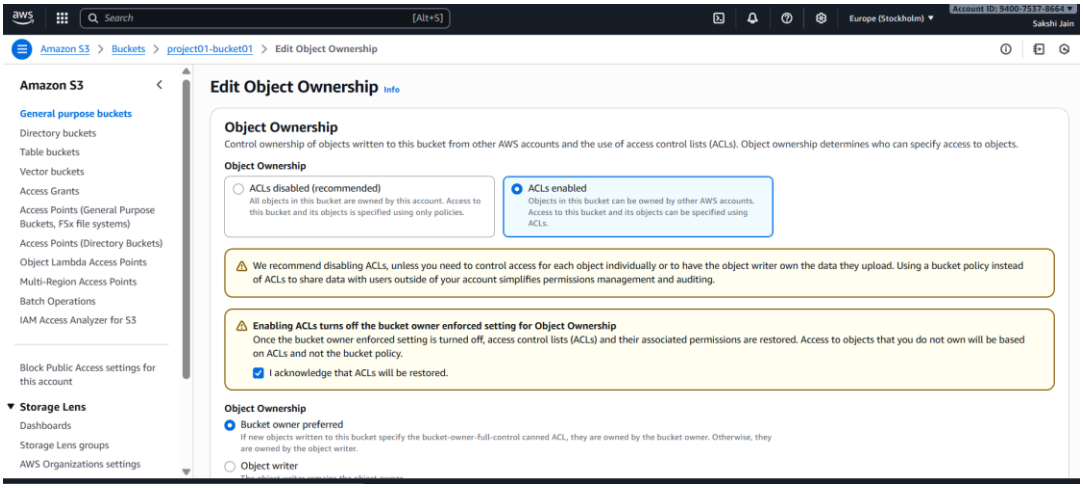
Set default and create bucket



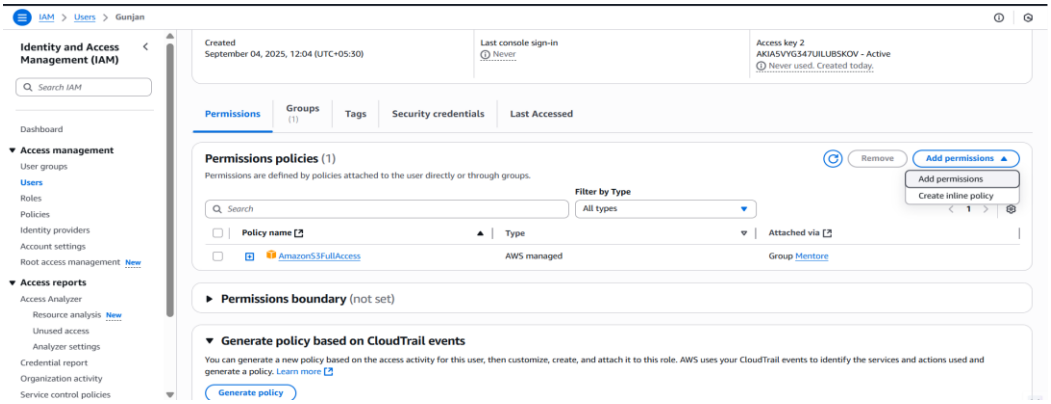
Allow S3 Bucket Publick access.

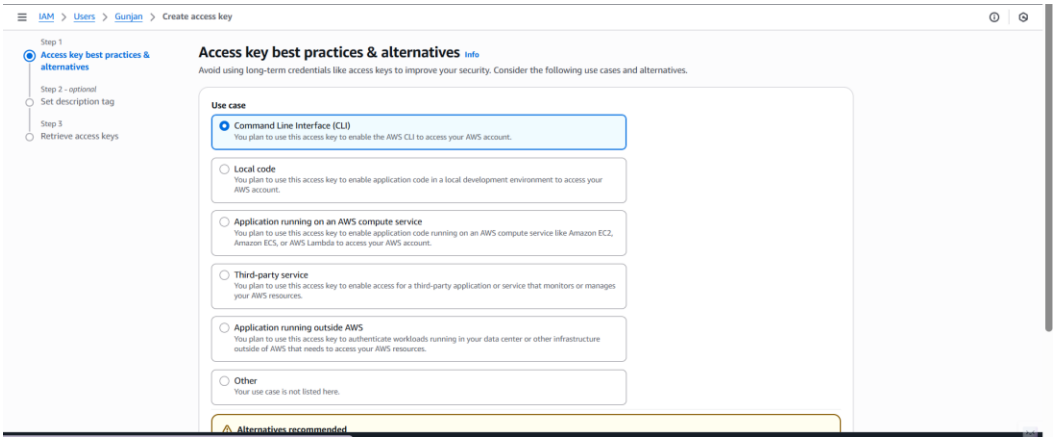


Set Object Ownership so that Object inside bucket would be public



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

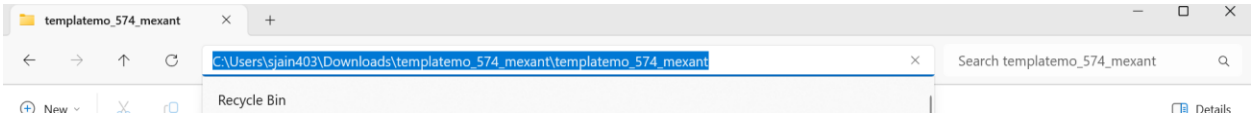
User your user access key here

```
C:\Users\sjain403>aws configure
AWS Access Key ID [****************CYUH]: AKIA5VYG347UILUBSKOV
AWS Secret Access Key [****************odJh]: xVylsW7QIzD9+nduTOHswvOUdwCxF6+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



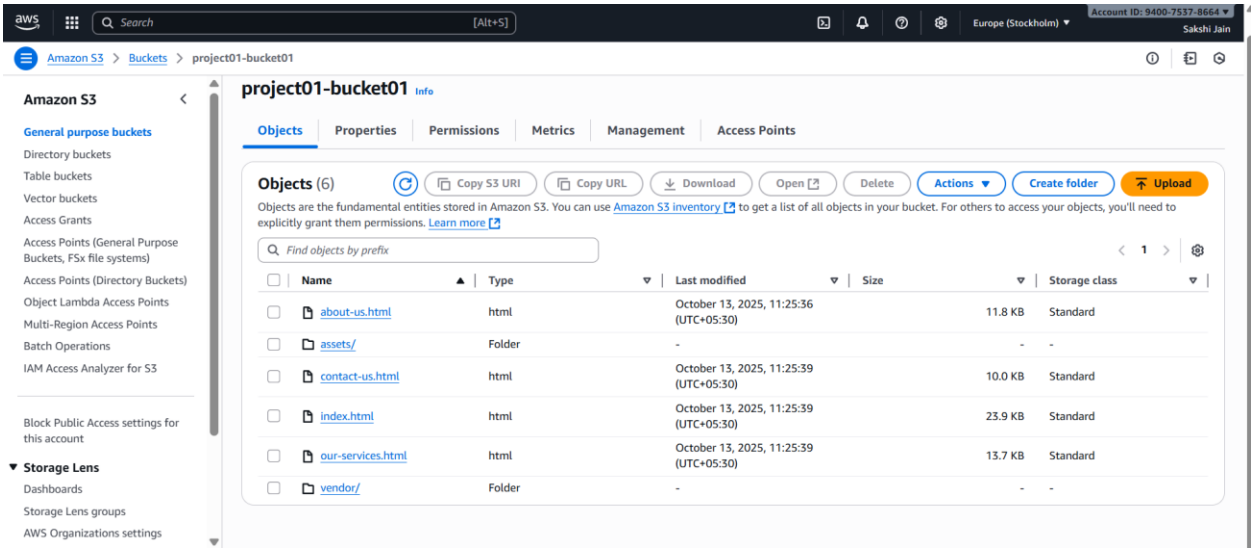
Now Goto command prompt and use command

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

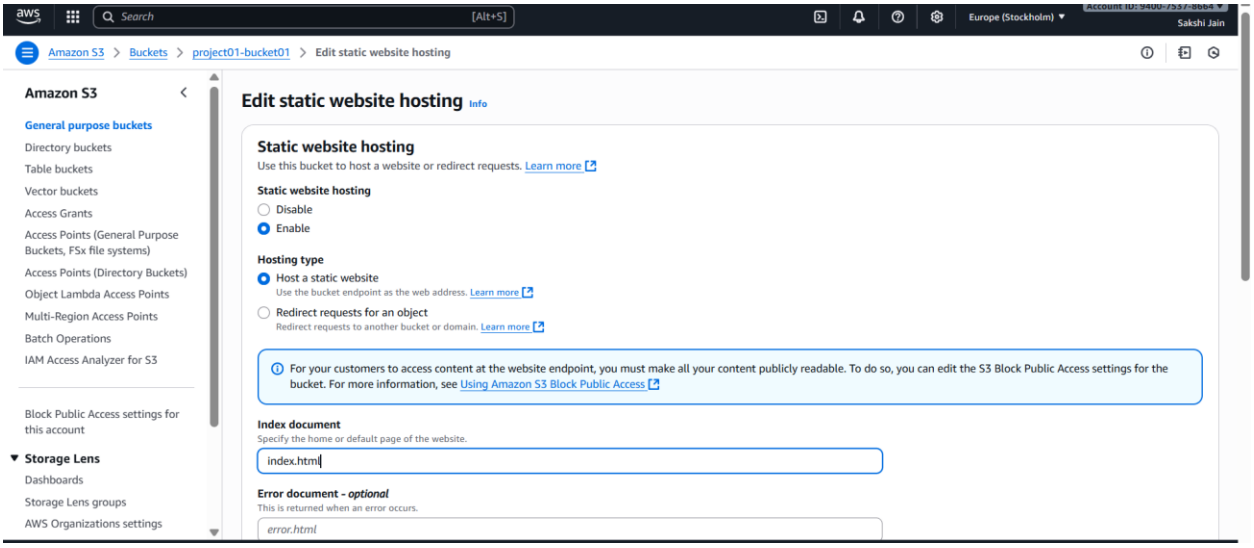
Upload Your Website Files via Windows CLI

```
C:\Users\sjain403>aws s3 sync "\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/
```

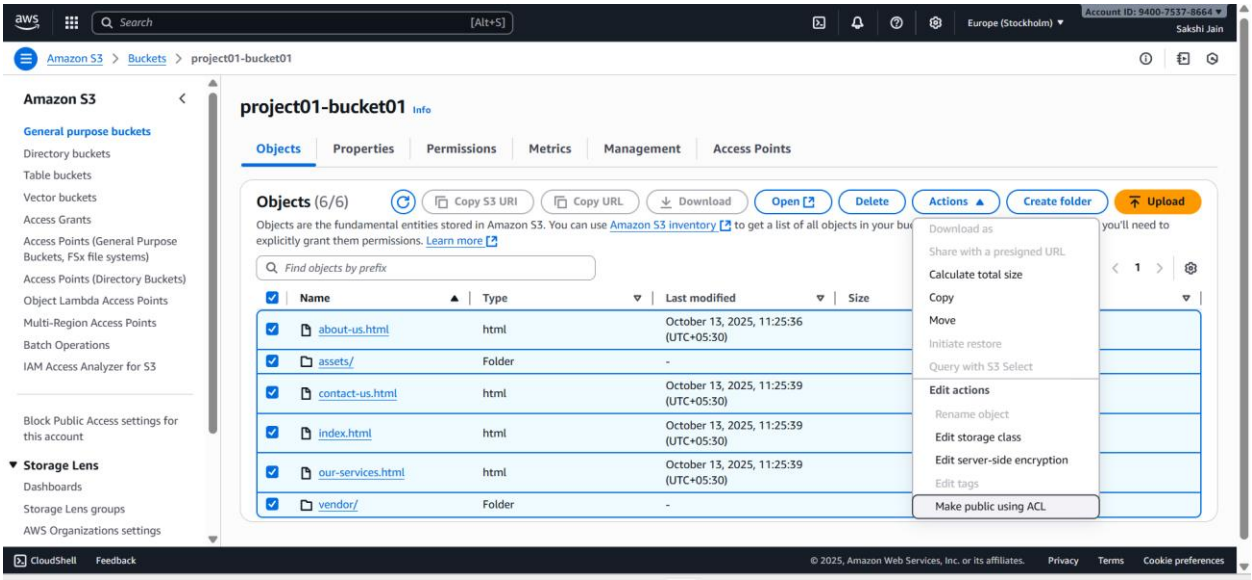
Step 5: Goto s3 and check you will upload data here.

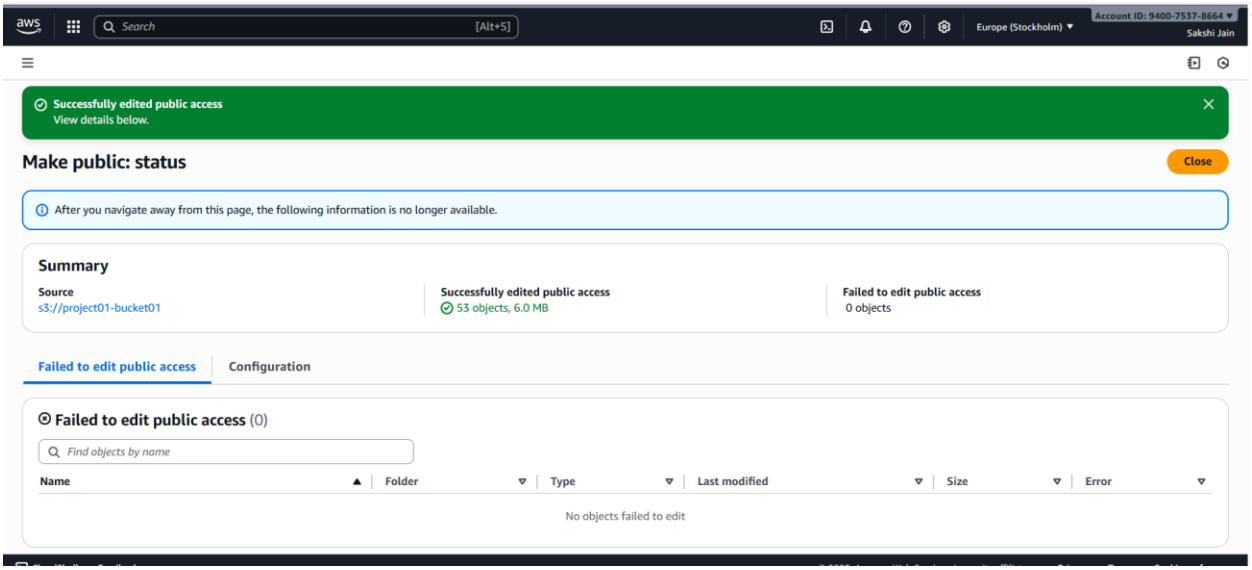


Now Goto properties and enable web hosting

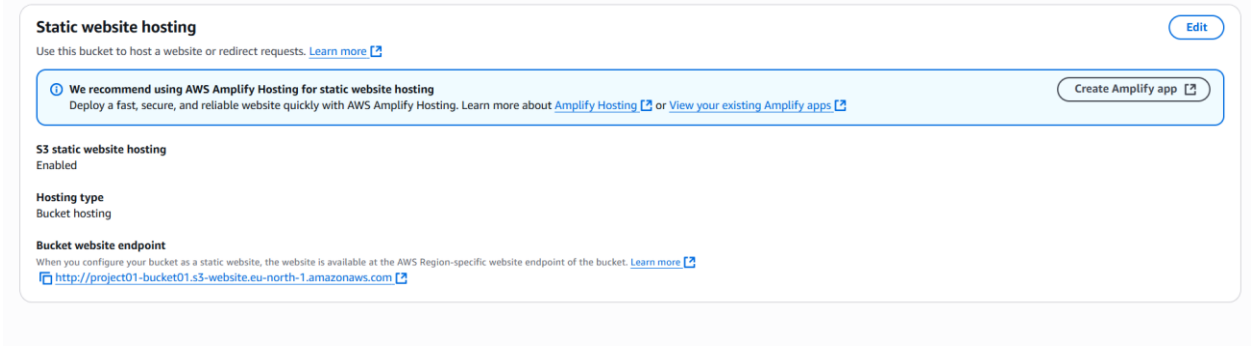


Make all object public using CLI

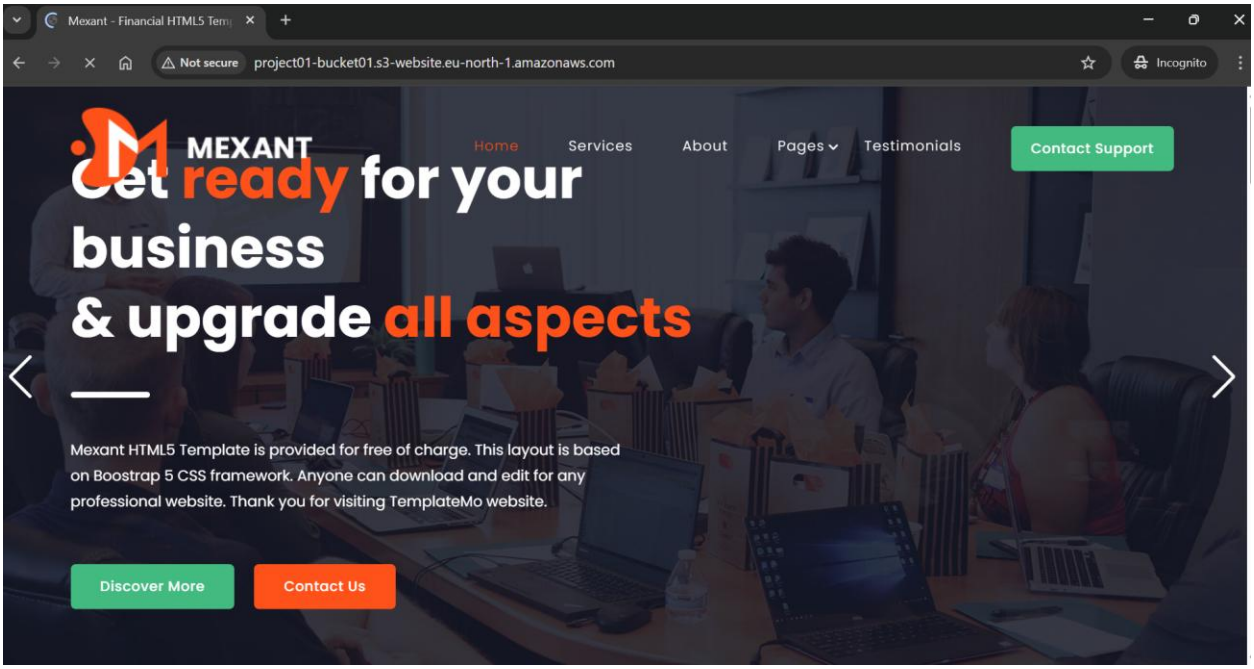




Step 6: Copy the website link http://project01-bucket01.s3-website.eu-north-1.amazonaws.com



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