# **Static Website Hosting on Amazon S3**

This project demonstrates how to host a static website using Amazon S3 with additional features like versioning, lifecycle policies, and cross-Region replication to enhance availability, durability, and cost efficiency. The website serves as an online presence for a café.

## **Table of Contents**

* Overview
* Technologies Used
* Setup and Configuration
  + Task 1: Extracting Website Files
  + Task 2: Creating an S3 Bucket for Hosting
  + Task 3: Uploading Content to the S3 Bucket
  + Task 4: Configuring Bucket Policy for Public Access
  + Task 5: Enabling Versioning
  + Task 6: Setting Lifecycle Policies
  + Task 7: Enabling Cross-Region Replication
* Summary

## **Overview**

In this project, an S3 bucket is configured to host a static website, with public access enabled. Advanced features like versioning, lifecycle management, and cross-Region replication were also added to improve functionality and durability.

## **Technologies Used**

* **Amazon S3**: Primary service for file storage and static website hosting.
* **IAM (Identity and Access Management)**: Used for setting permissions for cross-Region replication.

## **Setup and Configuration**

### **Task 1: Extracting Website Files**

1. **Objective**: Prepare the HTML, CSS, and image files for the website.
2. **Actions**:
   * Downloaded the .zip file with the website assets.
   * Extracted the files to obtain the index.html file and folders for CSS and images.

### **Task 2: Creating an S3 Bucket for Hosting**

1. **Objective**: Create an S3 bucket configured for static website hosting.
2. **Actions**:
   * Created an S3 bucket in **N. Virginia (us-east-1)** Region.
   * Disabled "Block all public access" to allow public access.
   * Enabled static website hosting, specifying index.html as the index document.

### **Task 3: Uploading Content to the S3 Bucket**

1. **Objective**: Upload the website files to S3 for hosting.
2. **Actions**:
   * Uploaded index.html, CSS, and images to the bucket.
   * Verified the setup by accessing the website through the S3 endpoint.

### **Task 4: Configuring Bucket Policy for Public Access**

1. **Objective**: Set up automatic public read access for all objects in the bucket.
2. **Actions**:
   * Configured a bucket policy to grant **read-only permissions** to public, anonymous users.
   * Verified that all uploaded objects are publicly accessible without additional configurations.

### **Task 5: Enabling Versioning**

1. **Objective**: Enable versioning to keep track of changes.
2. **Actions**:
   * Enabled versioning on the S3 bucket.
   * Made changes to index.html (updated bgcolor attributes) and uploaded the modified file.
   * Verified that previous versions were retained and accessible in the S3 console.

### **Task 6: Setting Lifecycle Policies**

1. **Objective**: Manage older versions to optimize storage costs.
2. **Actions**:
   * Configured **two separate lifecycle rules**:
     + **Rule 1**: Move older versions to **S3 Standard-Infrequent Access (IA)** after 30 days.
     + **Rule 2**: Delete older versions after 365 days.
   * These policies help control storage costs by automatically archiving and expiring outdated versions.

### **Task 7: Enabling Cross-Region Replication**

1. **Objective**: Set up cross-Region replication for redundancy and disaster recovery.
2. **Actions**:
   * Created a second S3 bucket in another AWS Region and enabled versioning on it.
   * Enabled cross-Region replication on the source bucket with the **CafeRole** IAM role for required permissions.
   * Configured the replication rule to replicate all objects from the source to the destination bucket.

## **Summary**

This project demonstrates how to host a static website on Amazon S3 with additional features:

* **Static Website Hosting**: Configured the S3 bucket to serve as a public static website.
* **Automated Public Access**: Applied a bucket policy for simplified, automated public access.
* **Version Control**: Enabled versioning to manage and track changes over time.
* **Lifecycle Management**: Set up lifecycle rules to archive and delete older content, optimizing storage costs.
* **Cross-Region Replication**: Configured cross-Region replication for improved redundancy and disaster recovery.

By implementing these configurations, this project showcases a robust and scalable approach to static website hosting on AWS.