

# AMAZON WEB SERVICES PROJECT-I

AIM: Create an S3 bucket, enable versioning on it, host a static website on the s3 bucket then make some changes on your website, reupload it and then rollback to your previous version.

## PROCEDURE:

We'll proceed step by step with proper understanding, documentation and demonstration wherever possible.

### **1. Creating a S3 Bucket:**

- Log in to your AWS Management Console.
- Navigate to the S3 service.
- Click the "Create bucket" button.
- Choose a unique bucket name and select a region.
- Click through the configuration options, leaving most settings at their defaults, and create the bucket.

The screenshot displays the AWS Management Console interface. At the top, the 'Create bucket' wizard is shown with the 'General configuration' tab selected. The 'Bucket name' field contains 'arpits-bucket-226' and the 'AWS Region' dropdown is set to 'Asia Pacific (Mumbai) ap-south-1'. Below this, a green notification bar states 'Successfully created bucket "arpits-bucket-226"'. The main console area shows the 'Buckets' list with one entry: 'arpits-bucket-226' in the 'Asia Pacific (Mumbai) ap-south-1' region, with 'Bucket and objects not public' access and a creation date of 'September 2, 2023, 13:40:51 (UTC+05:30)'.

**Create bucket** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

Bucket name

arpits-bucket-226

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

Asia Pacific (Mumbai) ap-south-1

Successfully created bucket "arpits-bucket-226"

To upload files and folders, or to configure additional bucket settings choose [View details](#).

[View details](#)

Amazon S3 > Buckets

**Account snapshot** [Learn more](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

**Buckets (1)** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Find buckets by name

Name	AWS Region	Access	Creation date
arpits-bucket-226	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	September 2, 2023, 13:40:51 (UTC+05:30)

So, we have successfully created a bucket named **arpits-bucket-226** and selected region as **Asia Pacific (Mumbai) ap-south-1** and leaving most of the configurations set as default. We'll change it later if required.

## 2. Enable Versioning:

- Select the newly created bucket in the S3 dashboard.
- Go to the "Properties" tab. Scroll down to the "Versioning" section.
- Click the "Edit" button.
- Select "Enable" and click "Save."

### Edit Bucket Versioning [Info](#)

#### Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☐ Suspend  
This suspends the creation of object versions for all operations but preserves any existing object versions.

☒ Enable

**i** After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.

Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

Cancel [Save changes](#)

### Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

Bucket Versioning

Enabled

Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

Here, we have enabled the bucket versioning and saved the changes made. Now, in the bucket versioning section it'll reflect **bucket versioning enabled** as shown in above picture.

Before heading forward, make sure that bucket is **publicly accessible**. For this follow the below steps:

- Go to "Permissions" tab.
- Select "Edit" in Block Public Access (Bucket settings).
- Disable the Block all public access button and save changes.
- Select "Edit" in Bucket Policy and copy down the code written (It is the bucket policy) as show in the figure below and save changes.

## 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 Public Access settings for this account are currently turned on

Block Public Access settings for this account that are enabled apply even if they are disabled for this bucket.

#### ☐ 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.

Go to **Bucket policy** and select “**edit**”. Now, copy the below code in the bucket policy section and write down your bucket name in the place where I underlined **orange** and save changes.

### Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Edit](#)[Delete](#)

#### Public access is blocked because Block Public Access settings are turned on for this account

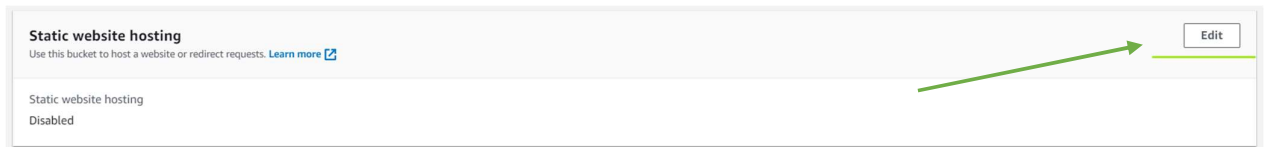
To determine which settings are turned on, check your [Block Public Access settings for this account](#). Learn more about [using Amazon S3 Block Public Access](#)

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "AllowPublicReadAccess",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": "s3:GetObject",  
9       "Resource": "arn:aws:s3::arpits-bucket-226/*"  
10    }  
11  ]  
12 }
```

After saving the above code, our bucket is publicly accessible.

### 3. Host a Static Website:

- In the bucket properties, go to the "Static website hosting" section.
- Click the "Edit" button. Select "Use this bucket to host a website."
- Specify the index document (e.g., index.html).
- Optionally, provide an error document. Here we'll only give index document.



### Edit static website hosting [Info](#)

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☒ Disable  
☐ Enable

### Edit static website hosting [Info](#)

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable  
☒ Enable

Hosting type

☒ Host a static website  
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object  
Redirect requests to another bucket or domain. [Learn more](#)

**i** 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](#)

Index document  
Specify the home or default page of the website.

index.html

Error document - *optional*  
This is returned when an error occurs.

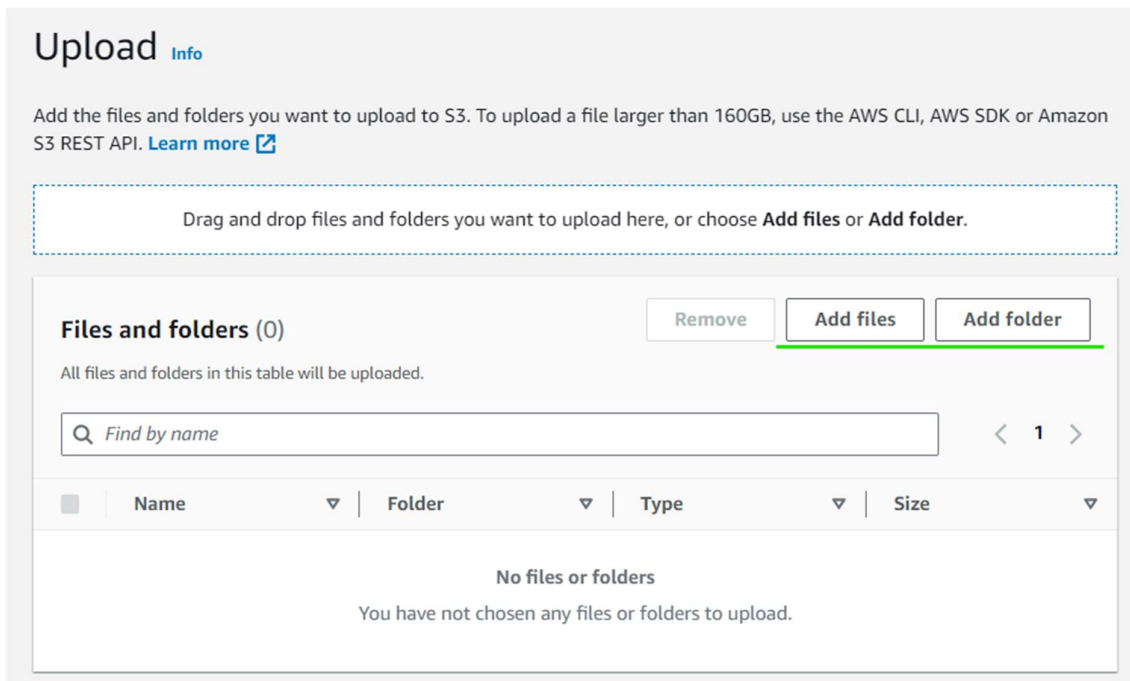
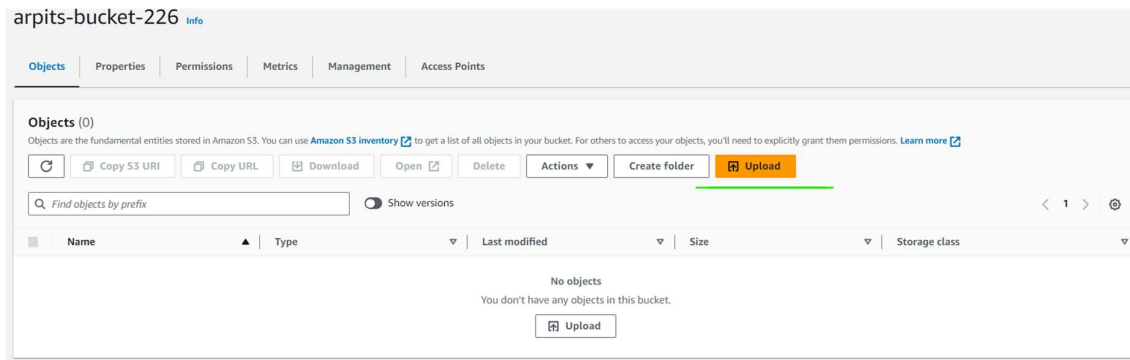
error.html

Redirection rules - *optional*  
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

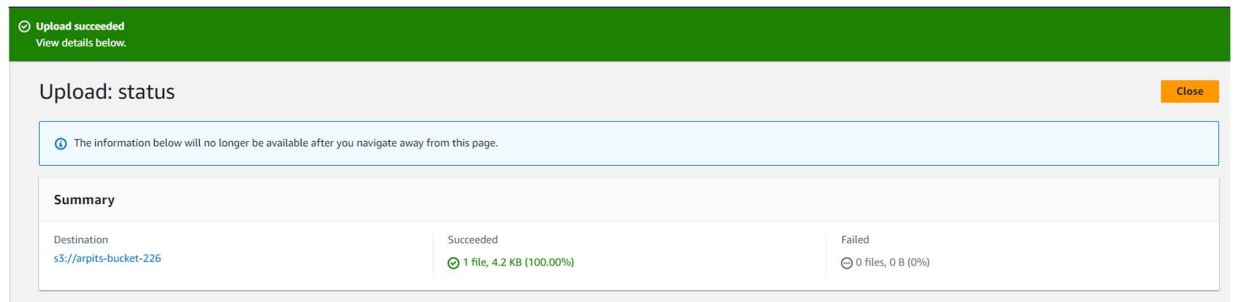
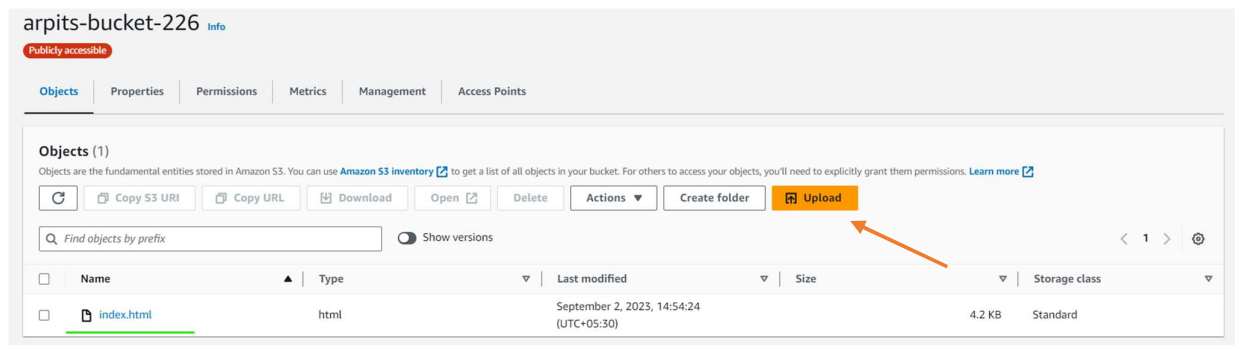
Here, we have enabled the **static website hosting** and hosting type as “**Host a static website**”. Moreover, we have specified the index document as **index.html** and error document as **error.html** and saved the changes made.

#### 4. Upload Your Website:

- Create a folder for your website files on your local machine if you haven't already.
- Upload your website files (HTML, CSS, JavaScript, etc.) to the S3 bucket created previously.

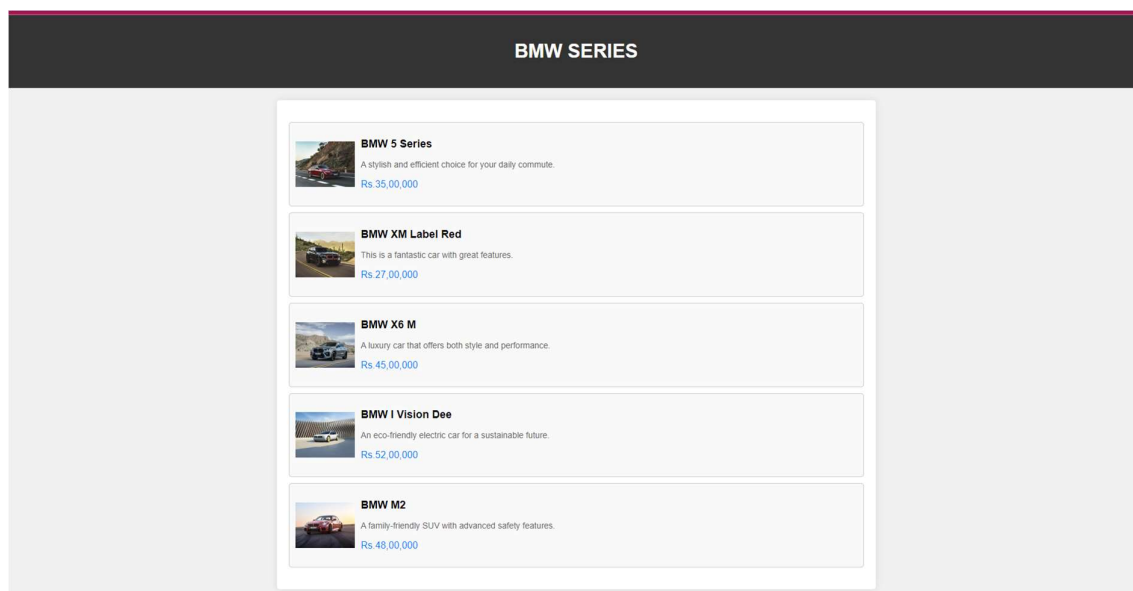


Add all the files/folders as required with the index.html file by selecting **Add files**, **Add folder** options and upload them.



We have successfully added the index.html file. In the status it will show **uploaded successfully** or **succeeded**.

Below screenshot shows that our website is working absolutely fine. It consists of different car models/series of the car BMW. There are 5 different models shown with their photographs, names, features and pricing.

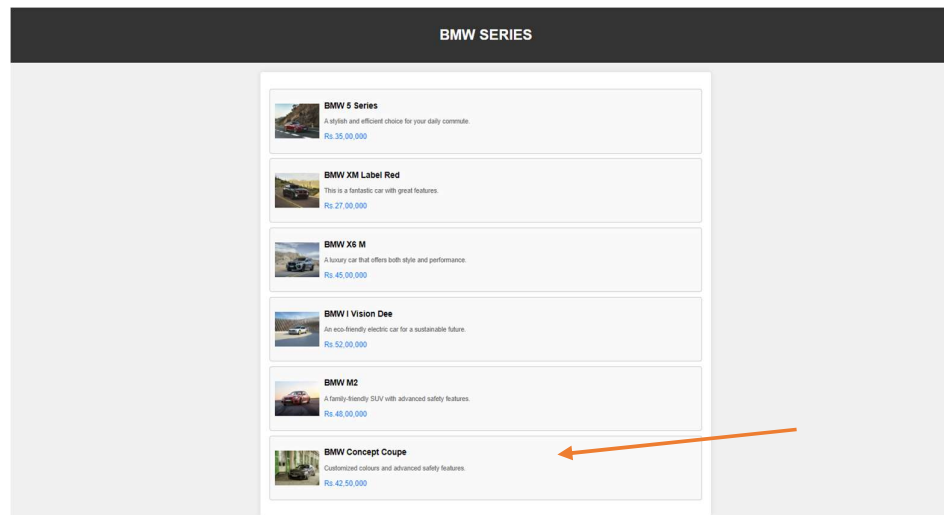


Now as asked, to make some changes in the files in our local machine and rollback to previous file, it will be done using **versioning**. Below are the later steps needed to do the asked.

## 5. Make Changes:

- Make the desired changes to your website files on your local machine.

Here we have added one more car (**BMW Concept Coupe**) in our list.



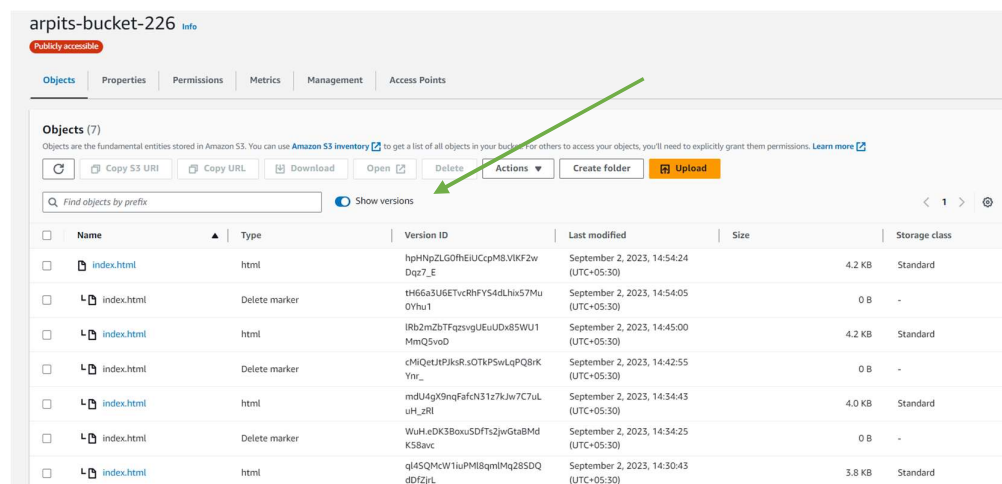
## 6. Reupload Changes:

- Upload the updated website files to the same S3 bucket, overwriting the existing files.

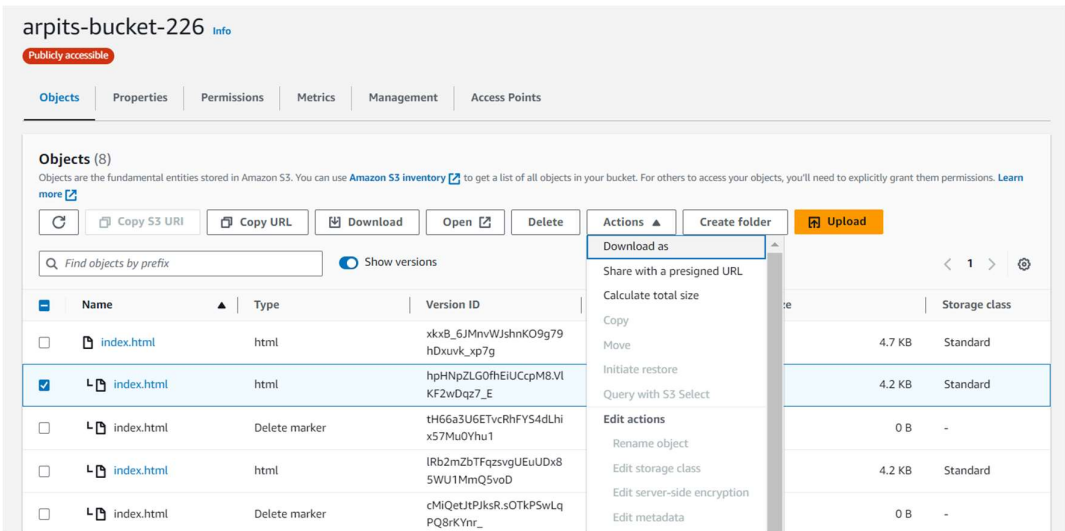
## 7. Rollback to Previous Version:

To roll back to a previous version of a file, you can do this through the S3 console:

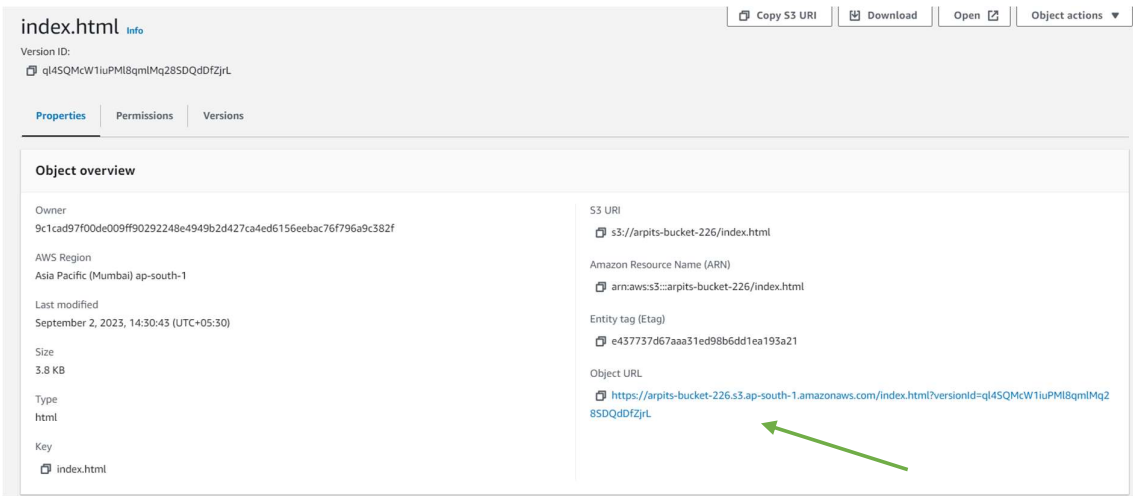
- In the **Buckets** list, choose the name of the bucket that contains the object.
- Choose **Versions**.
- Amazon S3 shows all the versions for the object.
- Select the check box next to the **Version ID** for the versions that you want to retrieve.
- Choose **Actions**, choose **Download**, and save the object.



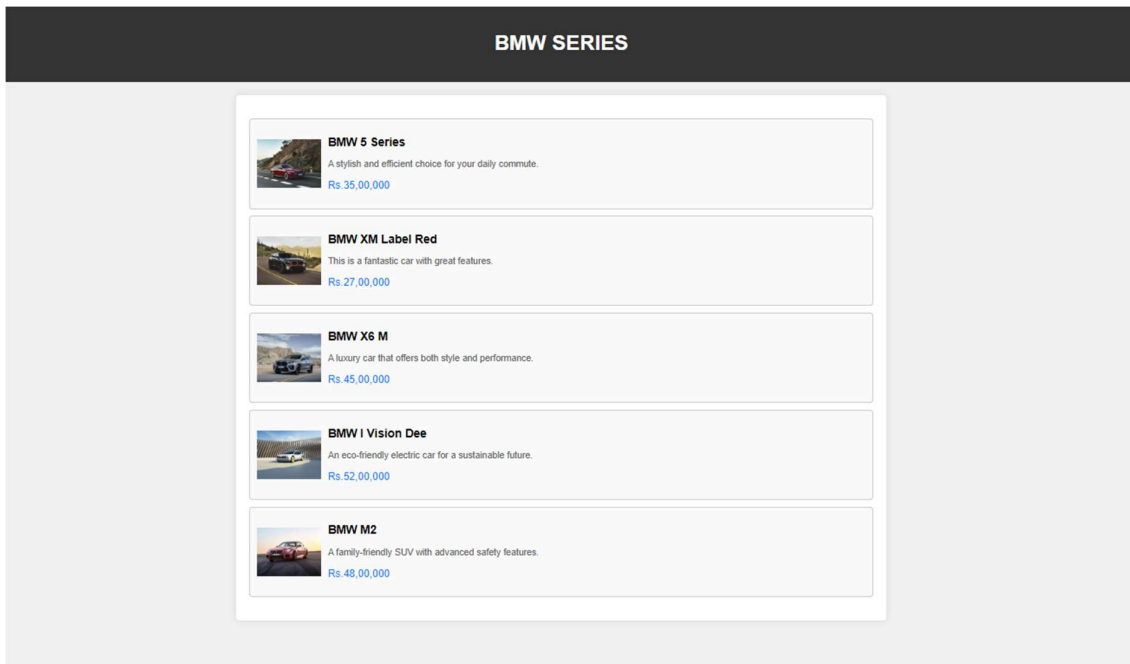
Select **Show versions**, a dropdown list will appear and select the version you want to rollback to.



Here, we have selected the just previous version of the bucket and downloaded it. And when opened it through **object URL**, we have got our previous version that we have created earlier.







Our earlier version with 5 models.

**RESULT:** Hence we have successfully created an S3 bucket, enabled versioning on it, hosted a static website on the s3 bucket then made some changes on your website, reuploaded it and then rollbacked to the previous version.

---