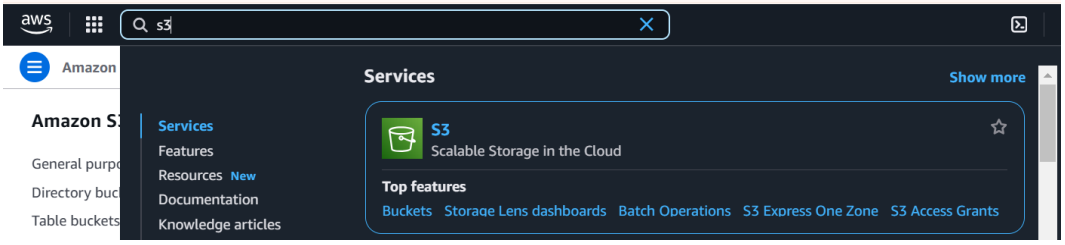
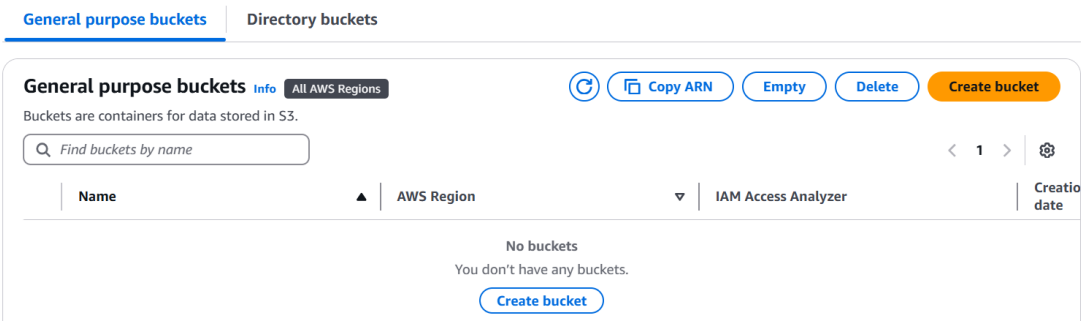
Project: 2 Scalable File Storage Solution

1. Create an S3 Bucket

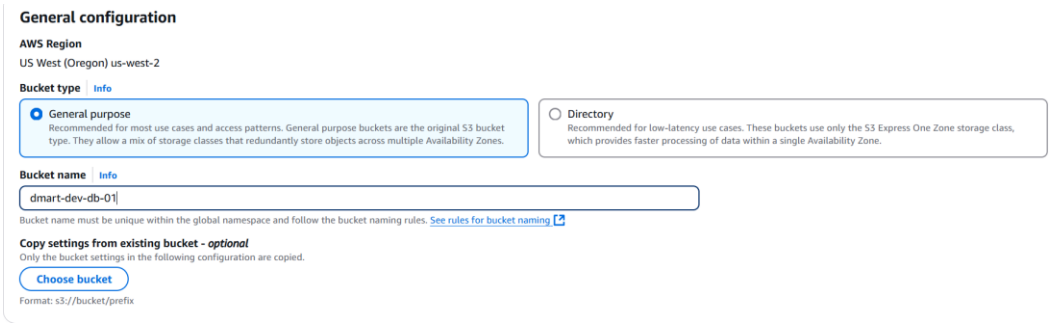
1. Open the AWS Management Console.



2. Navigate to **S3** and click **Create Bucket**.

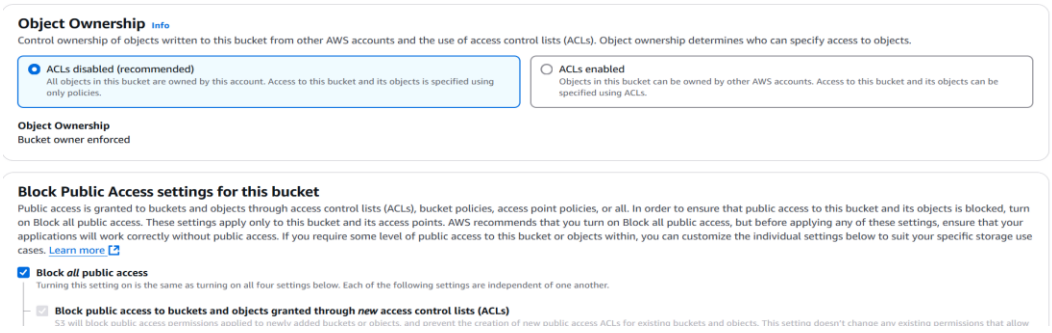


3. Provide a unique bucket name, e.g., dmart-dev-db-01.

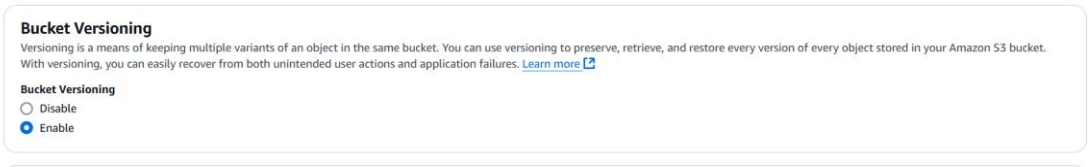


4. Select the region as per requirement.

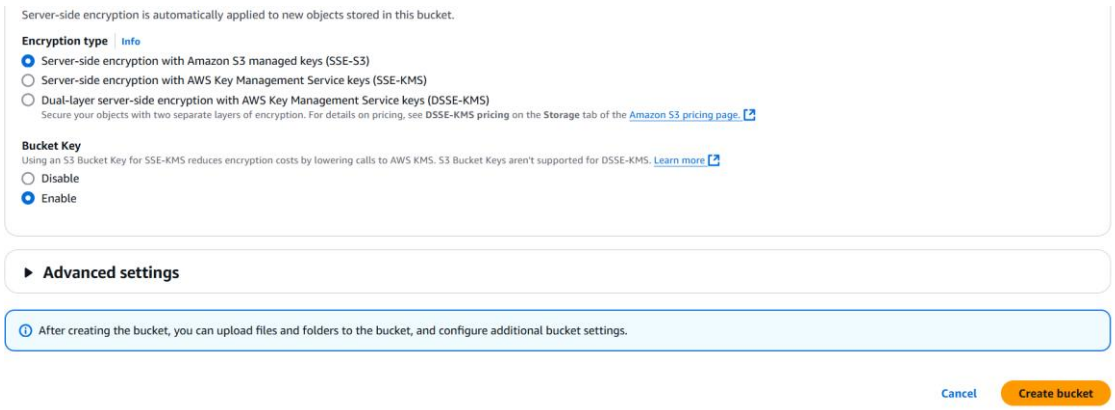
5. Under **Block Public Access settings**, select **Block all public access**.



6. Enable **Bucket Versioning** (useful for restoring files if deleted or modified).

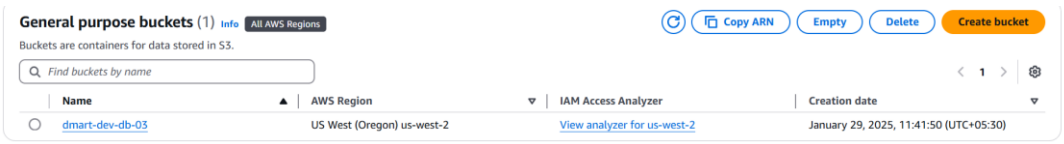


7. Click **Create Bucket**.

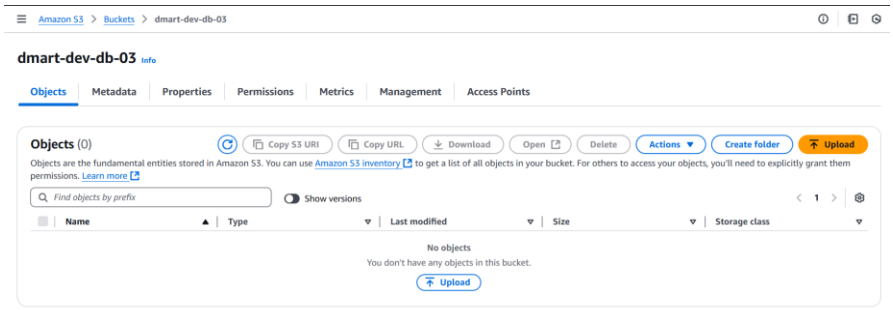


2. Upload Files to S3

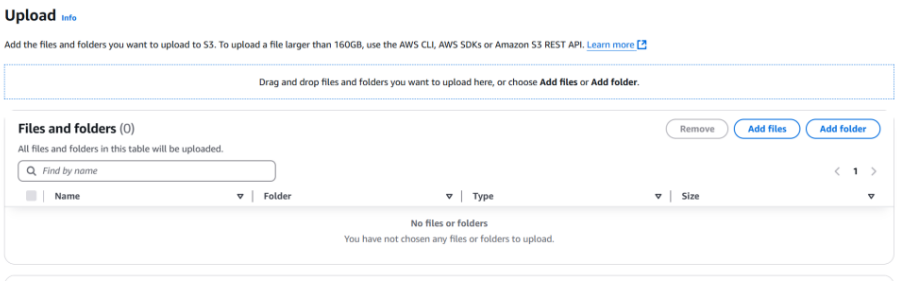
1. View the bucket created.



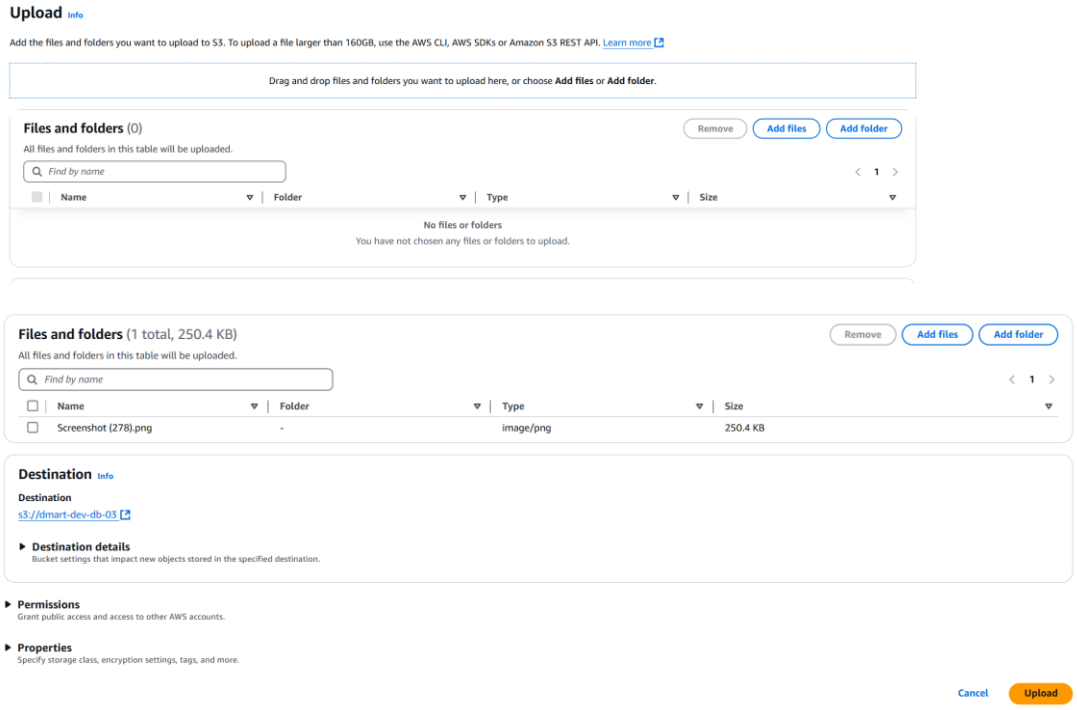
2. Open the created bucket.



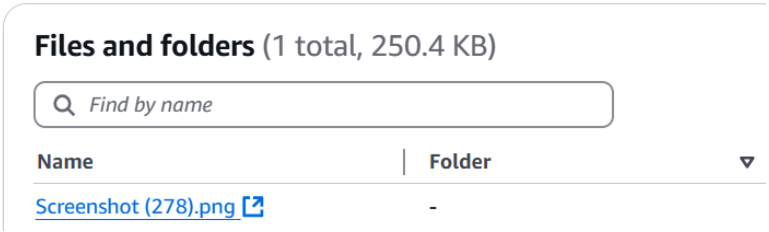
3. Click **Upload**.



4. Select files from your local system and click **Upload**.



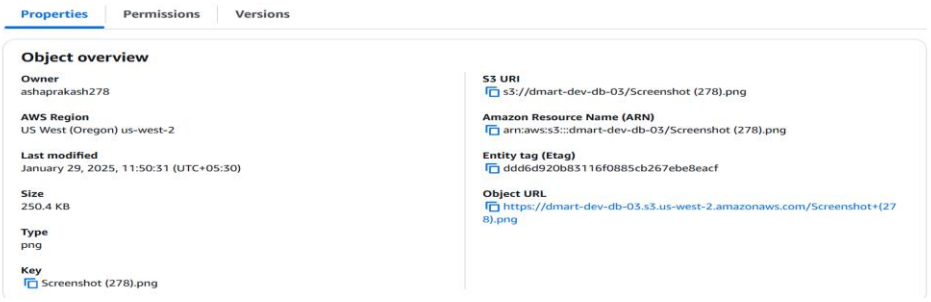
5. Verify that the files are uploaded successfully.



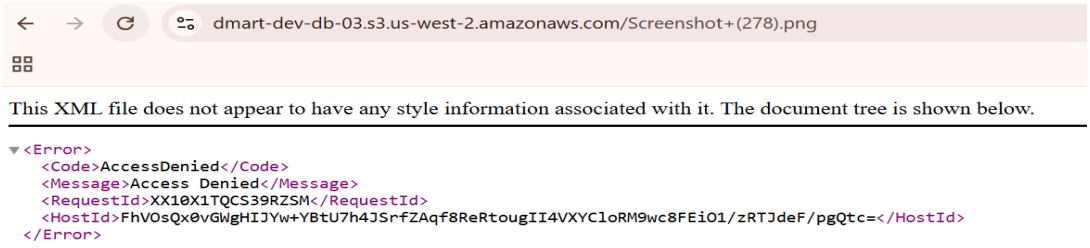
3. Grant Public Access to S3 Objects

1. Open the bucket and click on an uploaded file.
2. Copy the **Object URL** and paste it into the browser.

SnekaPrakash

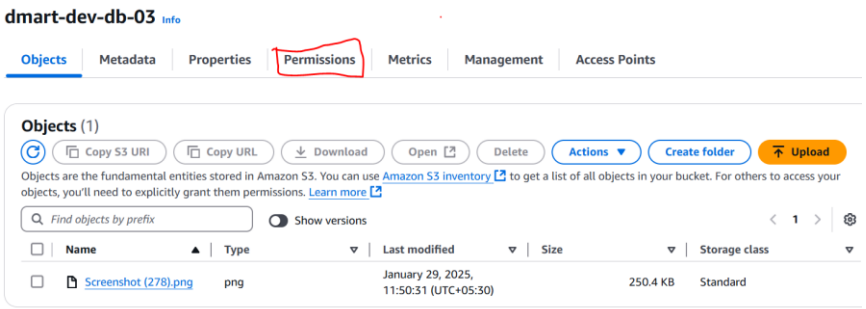


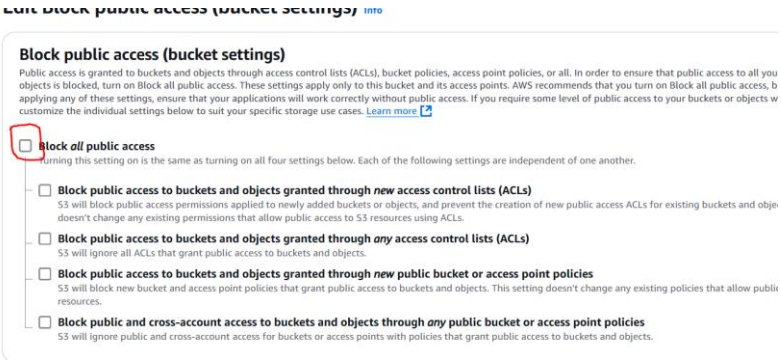
3. You will see an "Access Denied" error because public access is blocked.



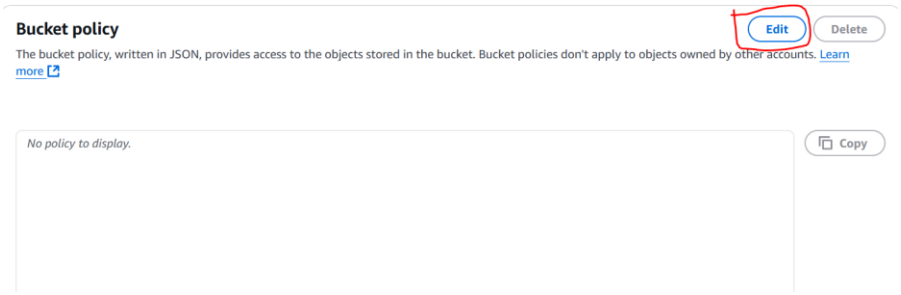
4. To allow public access:

   - Navigate to **Permissions** > **Bucket Policy**.

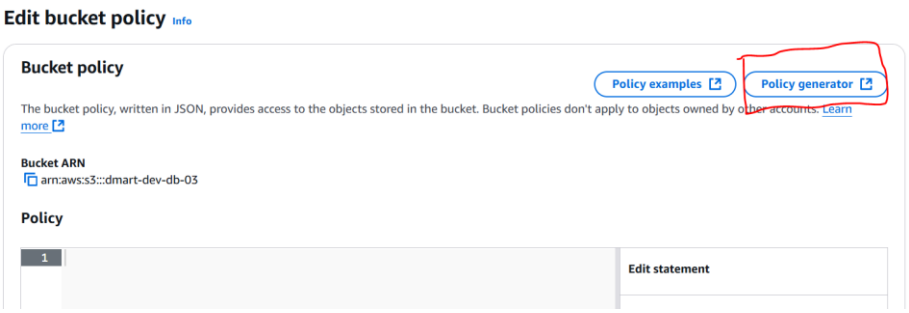




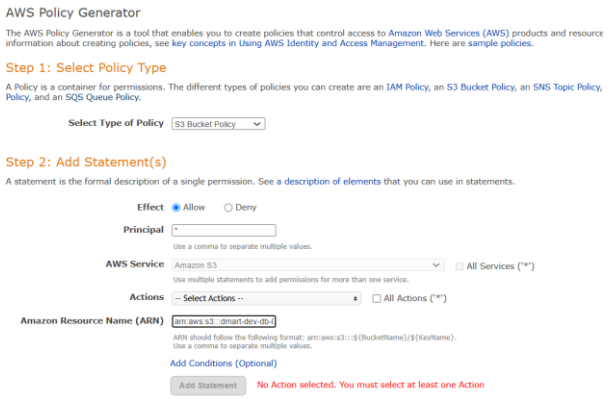
   - Click **Edit** and open **Policy Generator**.

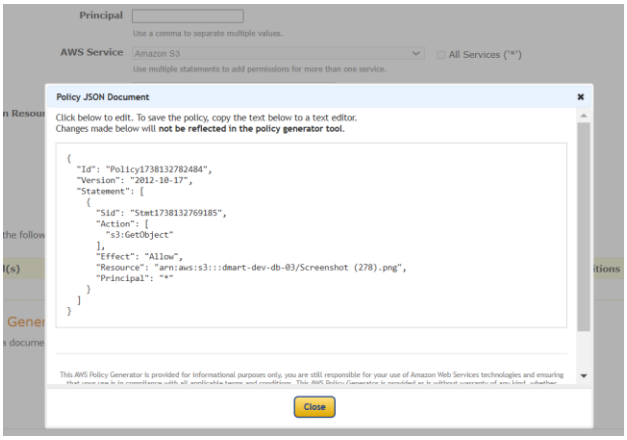


- Select **S3 Bucket Policy** and configure the policy for public read access.

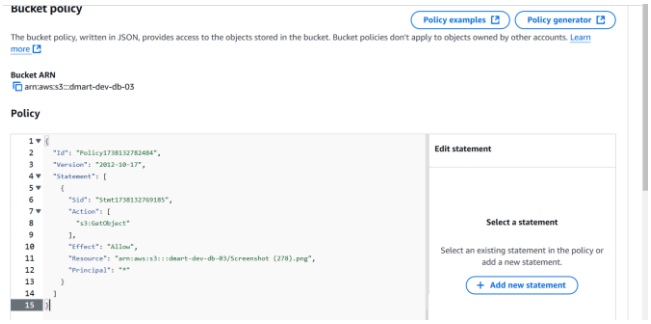


- Generate the policy and copy it.

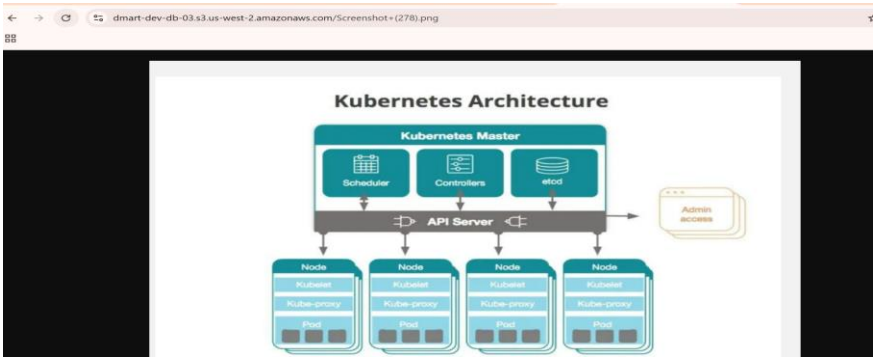




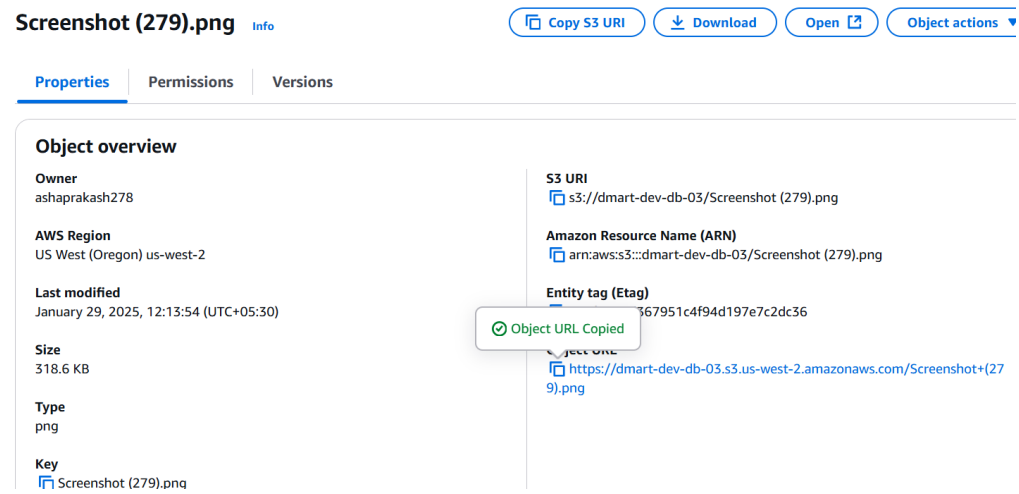
- Paste the policy into the **Bucket Policy** editor and save.

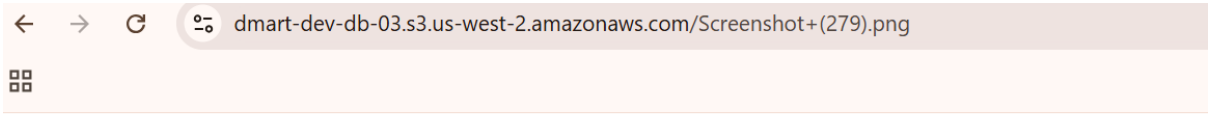


5. Now, accessing the file via the URL should be successful.



6. Upload another file and verify whether it is accessible (it will not be accessible until permissions are updated).





This XML file does not appear to have any style information associated with it. The document tree is shown below.

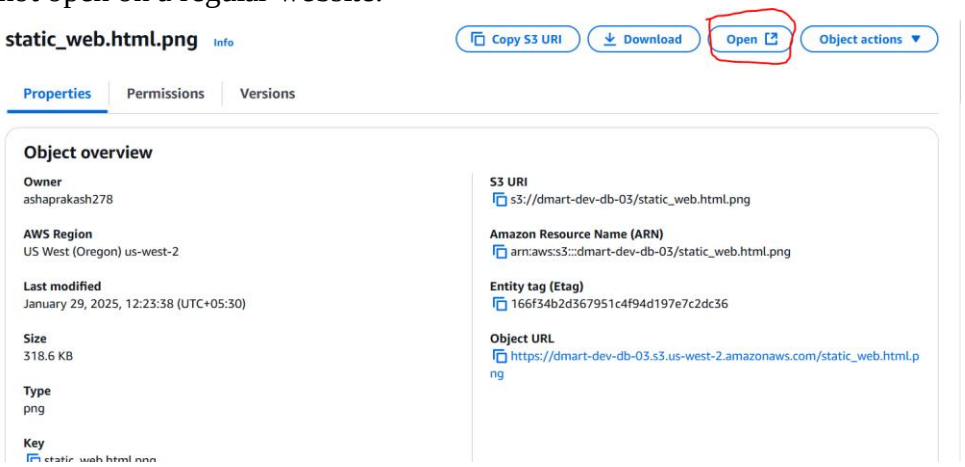
```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>9JJ3C3HWW93ME4FY</RequestId>
   <HostId>9Wrsdsvb1O+gQ7oZCPC45k5LAagCuEUeoX41f9hQuXQ3cj2itJrceIXwCZXrKLYSjQ3V87H9XKbmr1uha/5kPg==</HostId>
</Error>
```

Again, try to do the same process

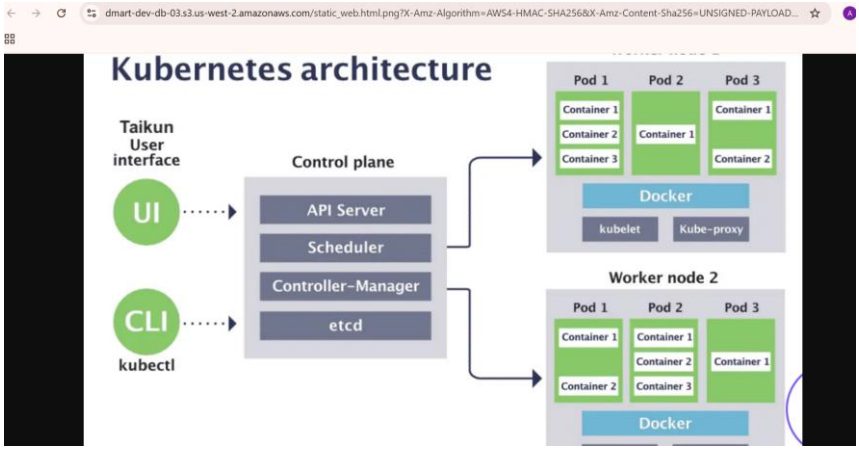
4. Enable S3 Static Website Hosting

1. Upload an index.html file to the bucket.

   Try to open it; it will open, but not on a website. Instead, it will open in AWS. It will not open on a regular website.

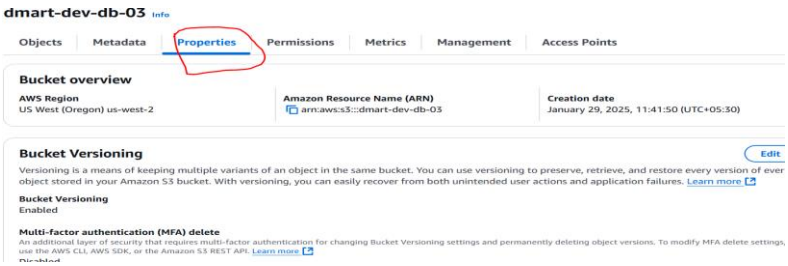


SnekaPrakash

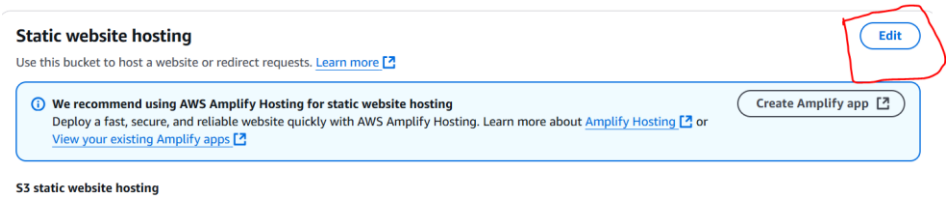


Opening through url:

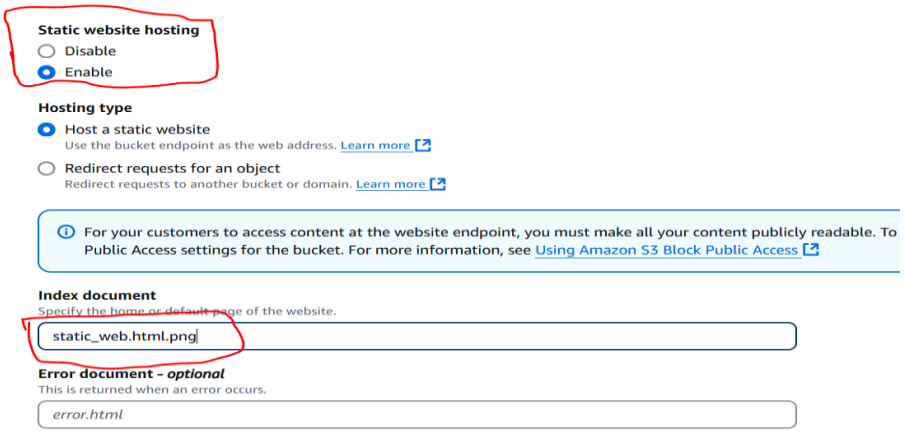
2. Enable **Static Website Hosting**:
    - Open the bucket and go to **Properties**.



    - Click **Edit** in the **Static Website Hosting** section.

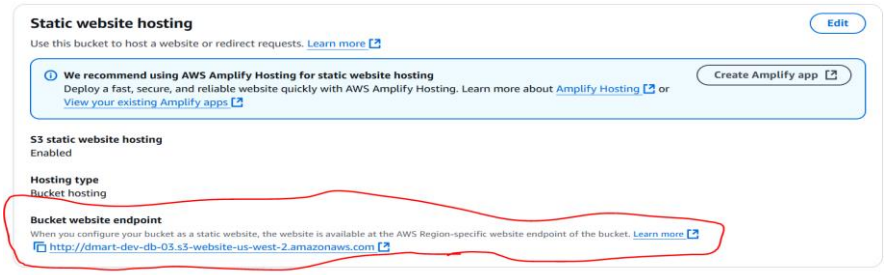


    - Enable it and specify **index.html** as the index document.

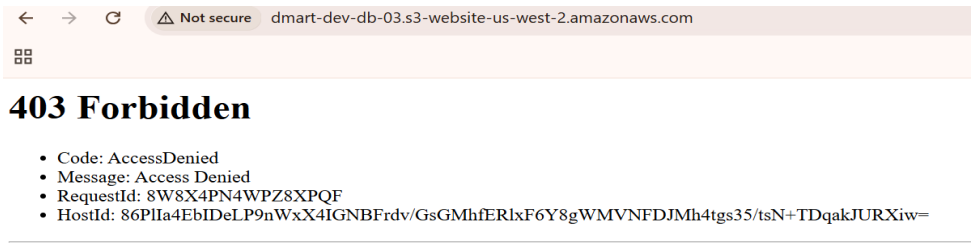


- Save the configuration.

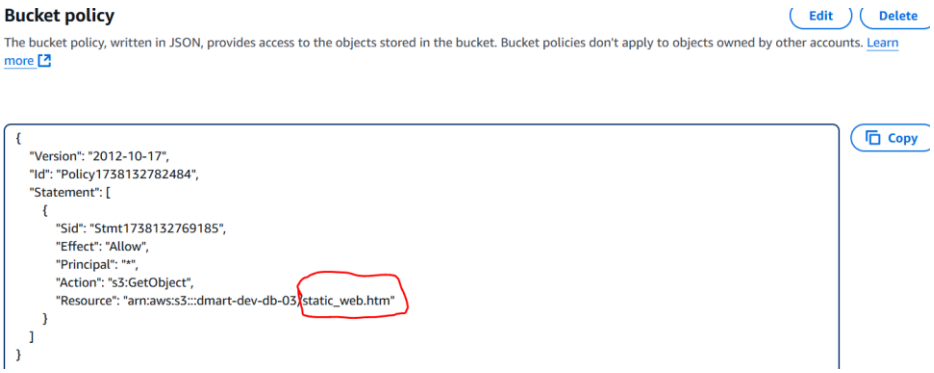
3. The index.html file will now open when accessed via the AWS S3 static website endpoint.



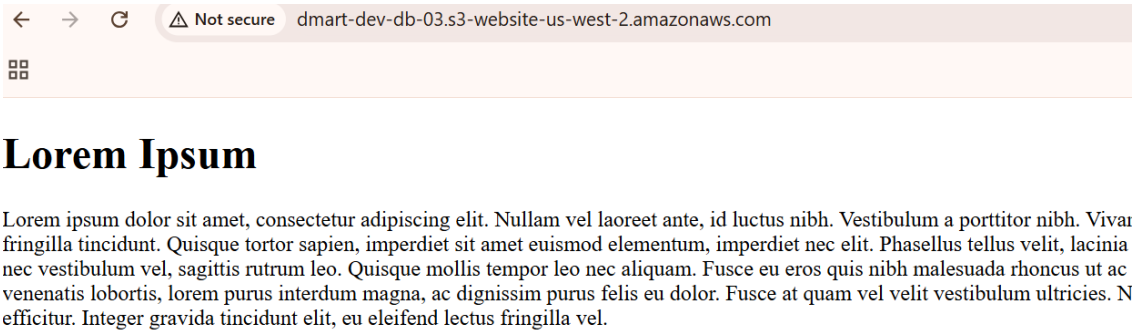
4. If you open the file directly in AWS, it will open, but it will not function as a website until public access is enabled and the correct policies are set.



5. Go to **Permissions** > **Bucket Policy** and add:

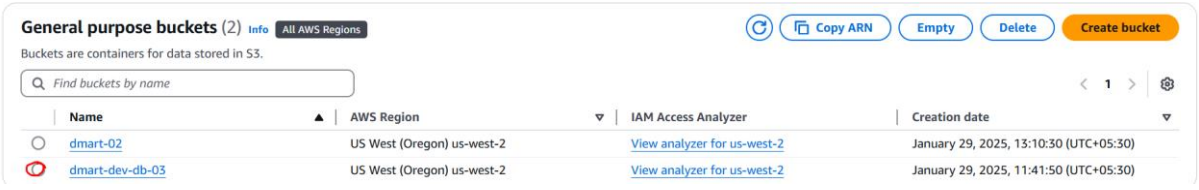


6. Save the changes.
7. The static website is now accessible via the provided S3 static website endpoint.

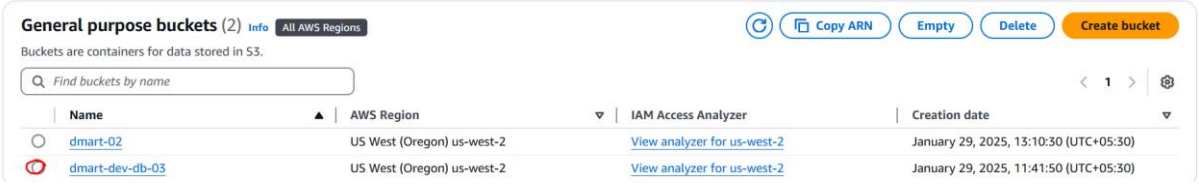


5. Configure S3 Replication for Backup

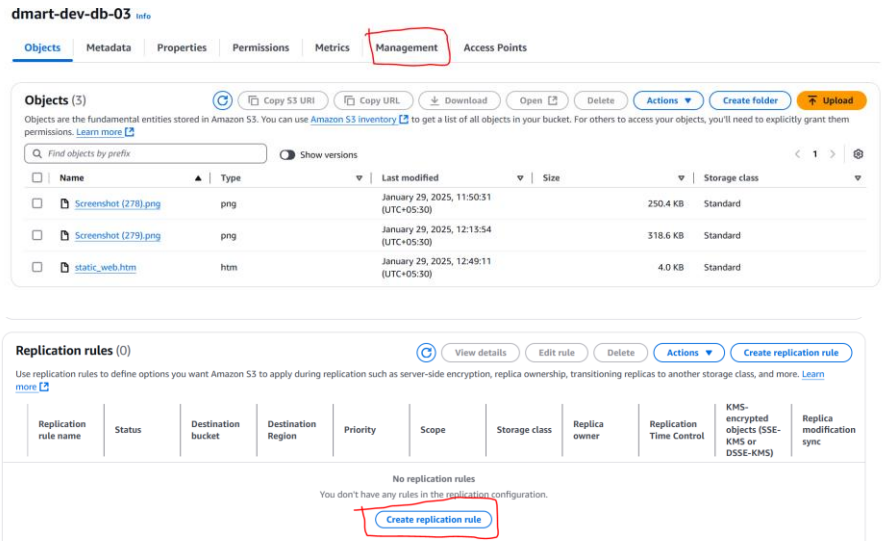
1. Create another bucket as a backup bucket.



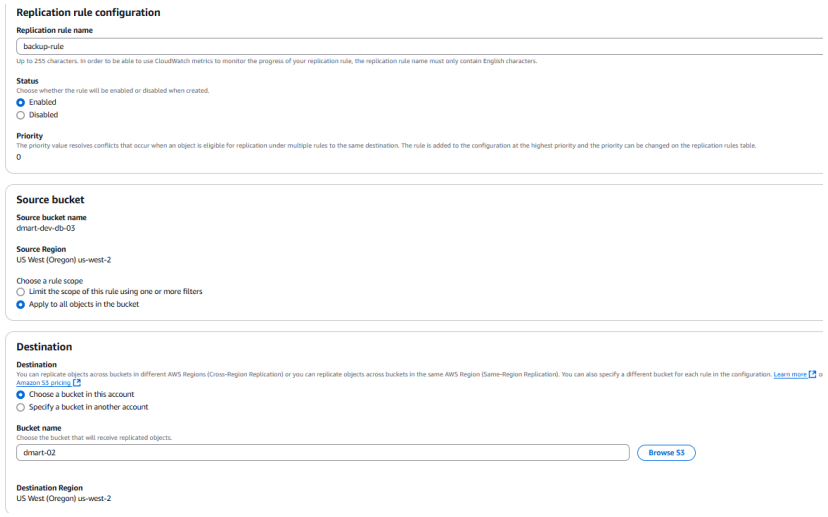
2. Open the main bucket (dmart-dev-db-01) and navigate to **Replication Rules**.



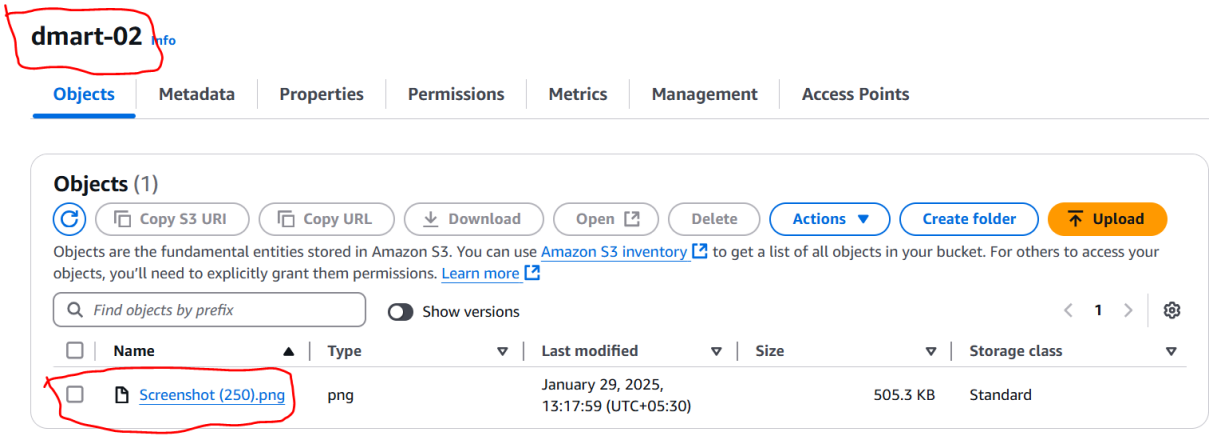
3. Create a **Replication Rule**:
    - Choose the backup bucket as the destination.



- Create a new IAM role to manage replication.



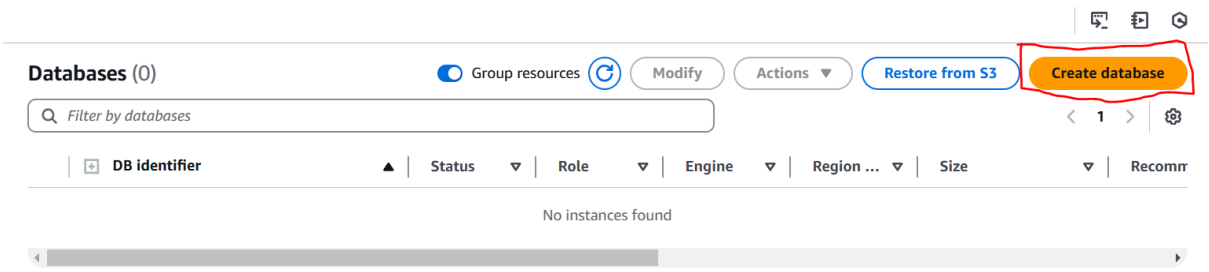
4. Save the rule.
5. Upload a new file to the main bucket and verify that it is automatically replicated in the backup bucket.



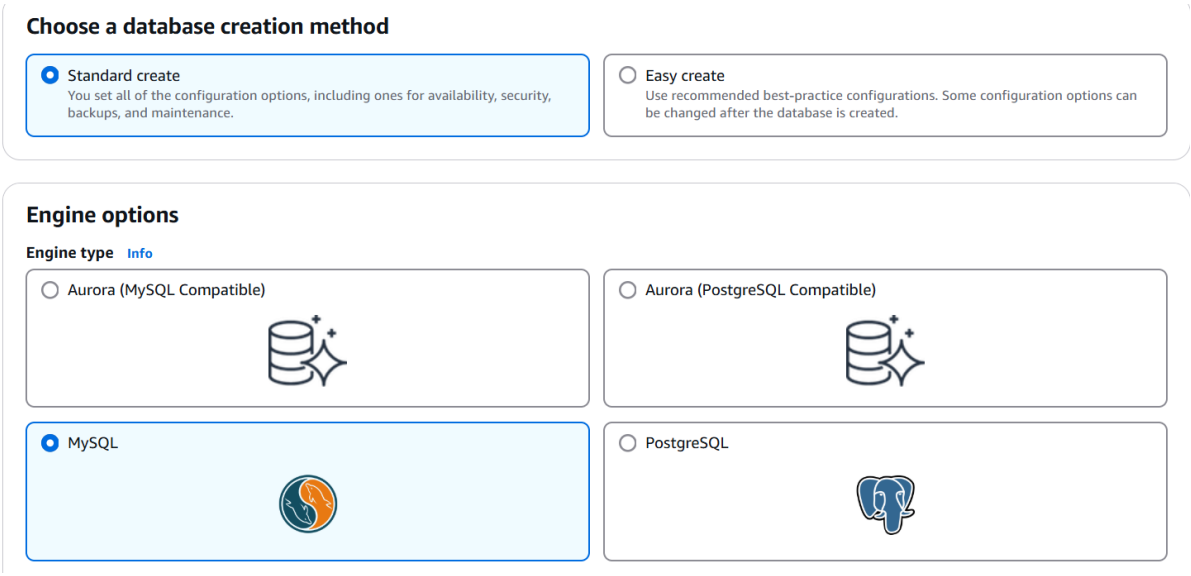
SnekaPrakash

6. Set Up RDS (MySQL Database)

1. Open AWS and search for **RDS**.
2. Click **Create Database**.



3. Select **MySQL** as the engine type.



4. Choose **Free Tier** template.

5. Set:
   - **Username:** admin
   - **Password:** admin@123
6. Configure storage:
   - **Allocated Storage:** 500GB
   - **Maximum Storage:** 1000GB
7. Enable **Public Access**.

SnekaPrakash

8. Click **Create Database**.

7. Configure Security Group for MySQL Access

1. Navigate to **EC2** > **Security Groups**.
2. Locate the security group associated with the RDS instance.
3. Edit inbound rules to allow MySQL (port 3306) from your IP or a specific range.

Conclusion

This process establishes a **Scalable File Storage Solution** by leveraging AWS S3 for storage, enabling replication for backup, and setting up an RDS database for managing structured data. The system ensures high availability and scalability while maintaining security and controlled access.