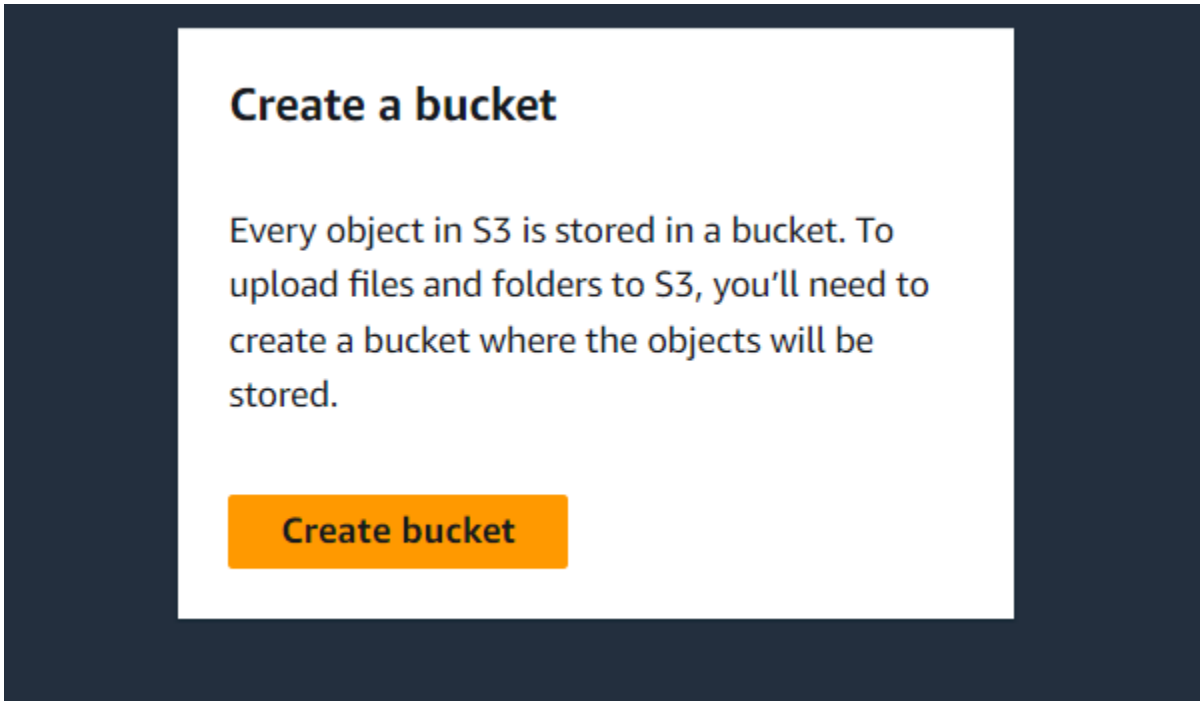
Basic labs:

2. **S3 Storage Fundamentals Lab**
    - **Objective**: To gain hands-on experience with Amazon S3 by performing basic storage operations.
    - **Approach**: This lab involves creating an S3 bucket, uploading files to it, and setting up bucket policies for access control. Students will explore the S3 management console, learn about object storage, and understand the concepts of buckets and objects.
    - **Goal**: Students will understand how to use S3 for storing and managing data, learn about S3 security and permissions, and become familiar with S3's user interface.

Steps Involved

1) **Create S3 bucket**



2) **Provide General Configuration**

General configuration

AWS Region

US East (N. Virginia) us-east-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 - *New*
  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

first-bucket-sonu

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

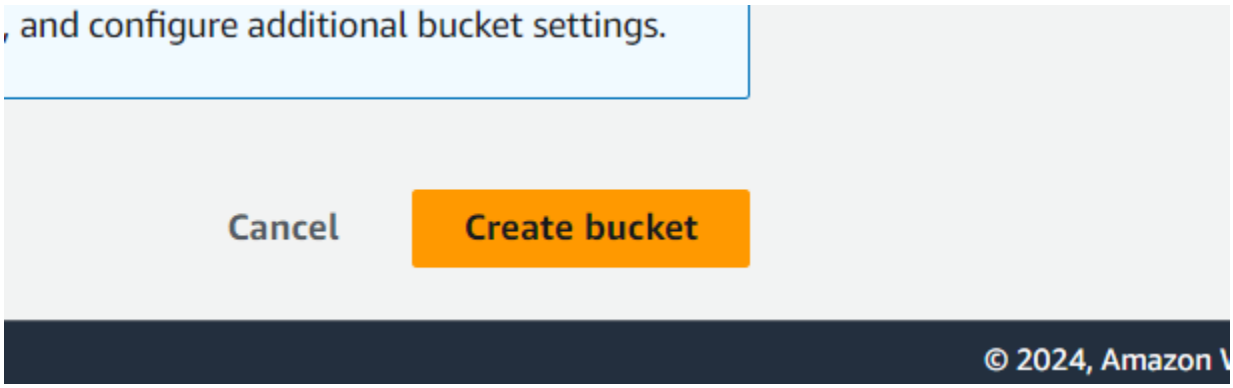
3) Block all public access

- [x] **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.
  - [x] **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.
  - [x] **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.
  - [x] **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.
  - [x] **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.

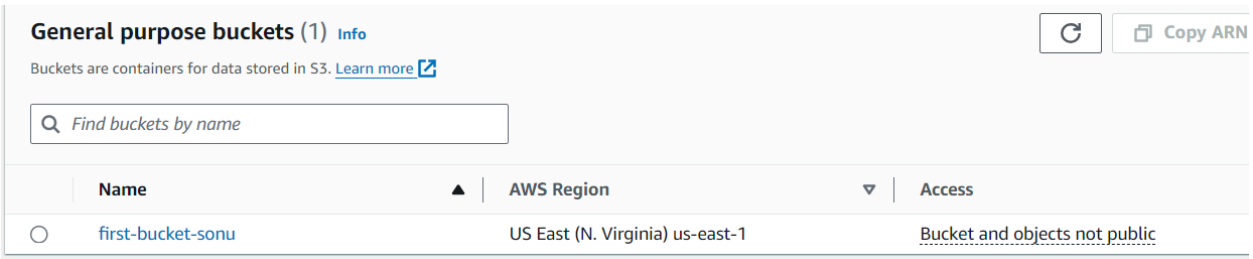
Bucket Versioning

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

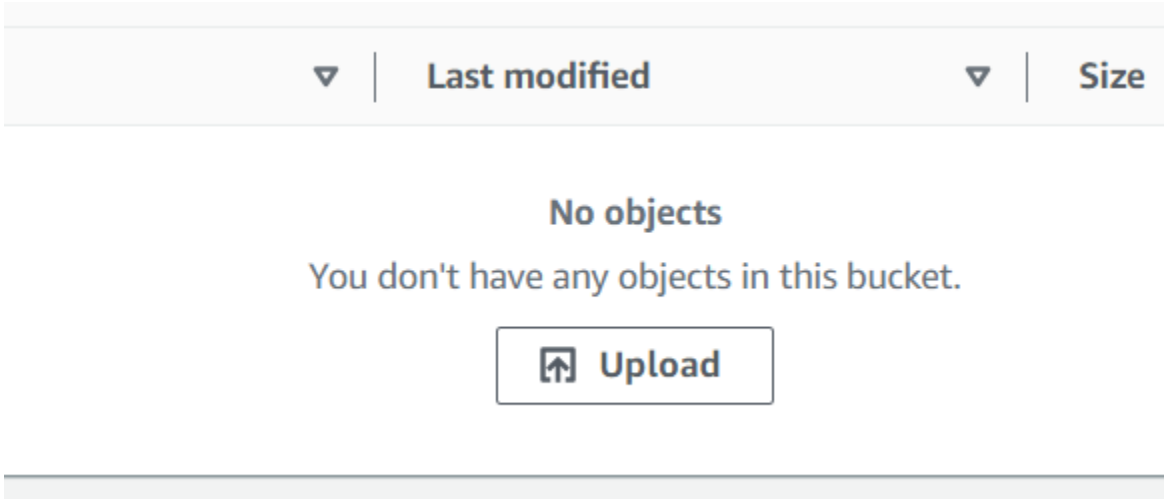
4) **Click on Create Bucket**



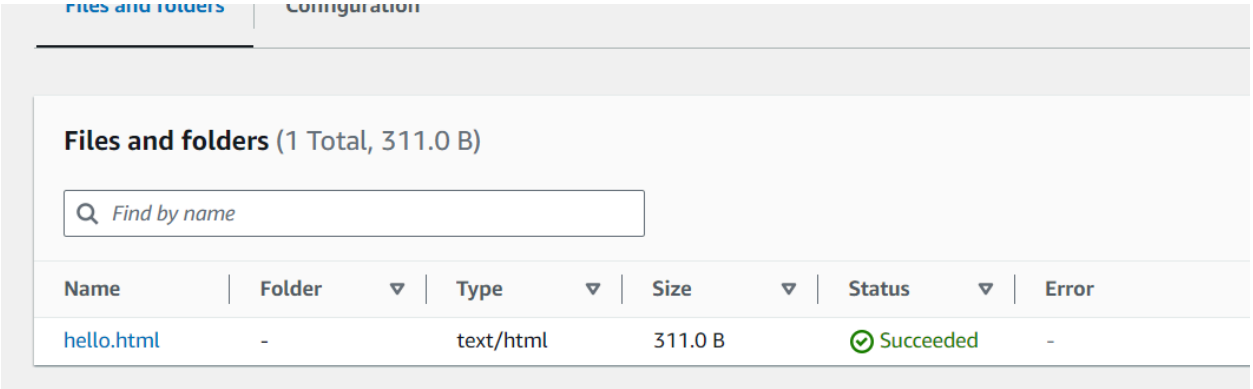
5) **The bucket is then created.**



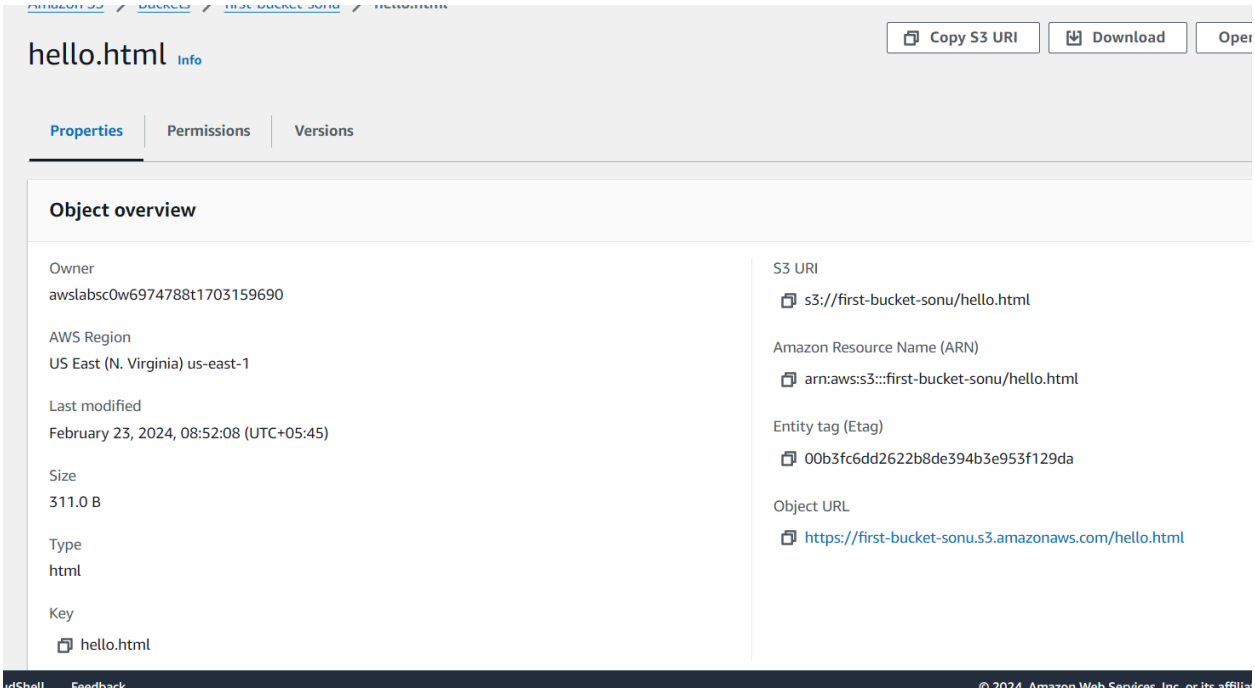
6) **Click on Upload button and upload your file.**



7) **The file has been uploaded.**



8) **Click on Object URL.The link redirects to the new tab. The errors is shown as below. When the file is viewed from object url link, the permission accessed denied is shown, as the "Block all public access" setting was checked during file upload.**



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>EZEH1J1KPWFWGFY3</RequestId>
   <HostId>HgiQp1PLwXldajygpJl8nhaRoyLSd4JAvW5gUaCUco5gHMNjStlgHH5WIupoLgn7egKO0IE8rb0=</HostId>
 </Error>
```


But if we click on Open button, it shows no error.

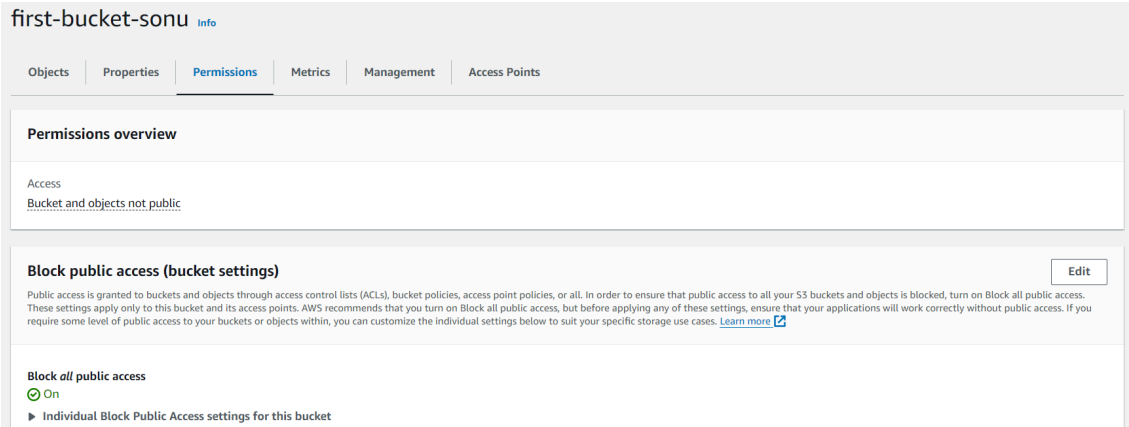
first-bucket-sonu.s3.us-east-1.amazonaws.com/hello.html?response-content-disposit

Hello ,This is Amazon Web services

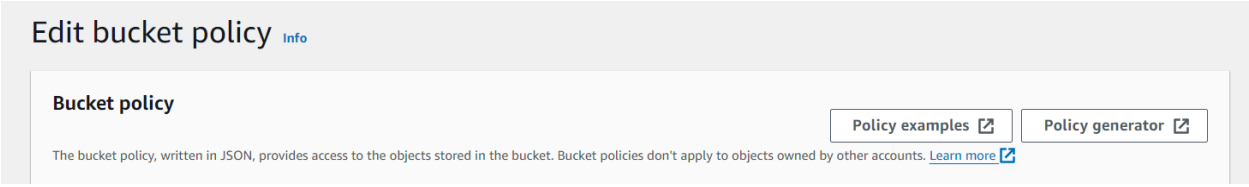
Lets get started with AWS.

Setting up bucket policies for access control

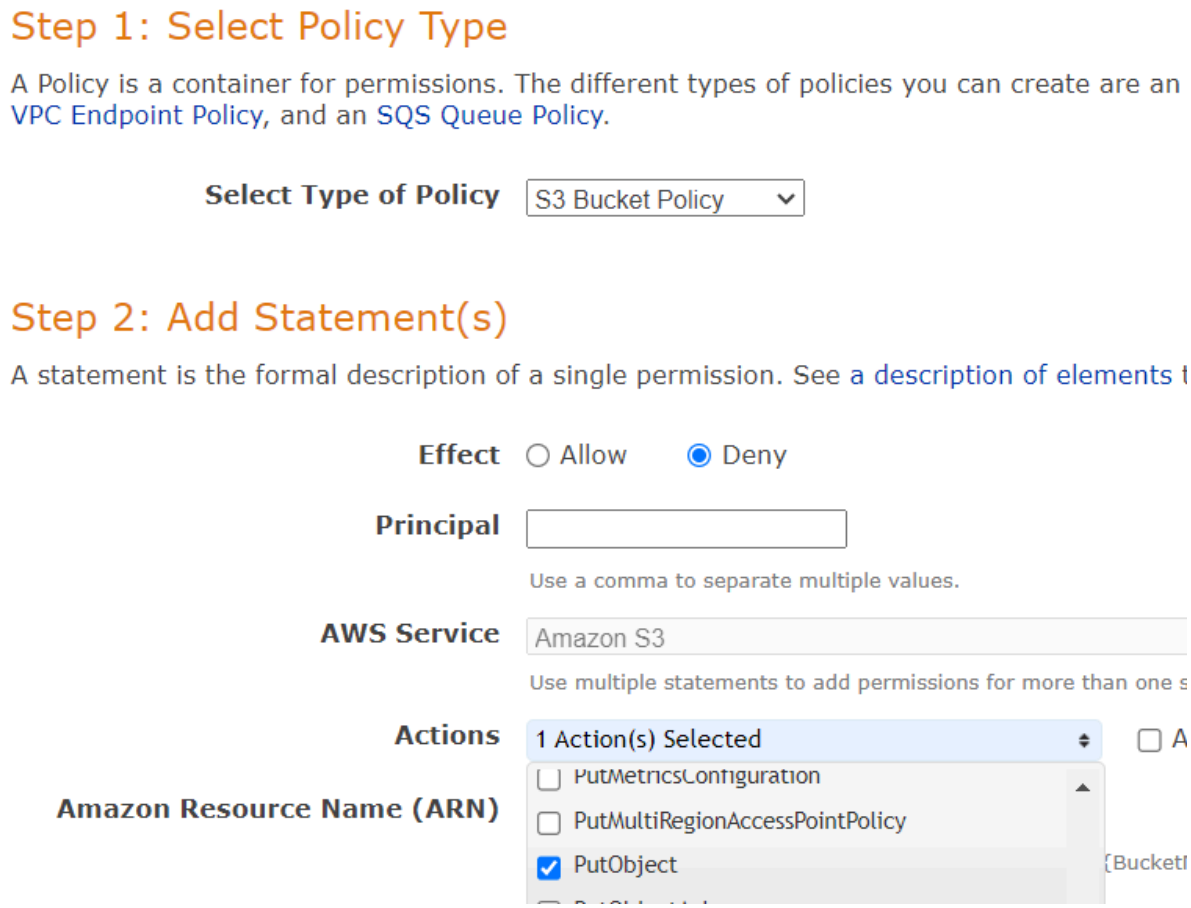
9) **Select “permissions” tab and “edit” button of “Bucket policy” section to edit the policies of the bucket, which will take to new screen of bucket policy.**



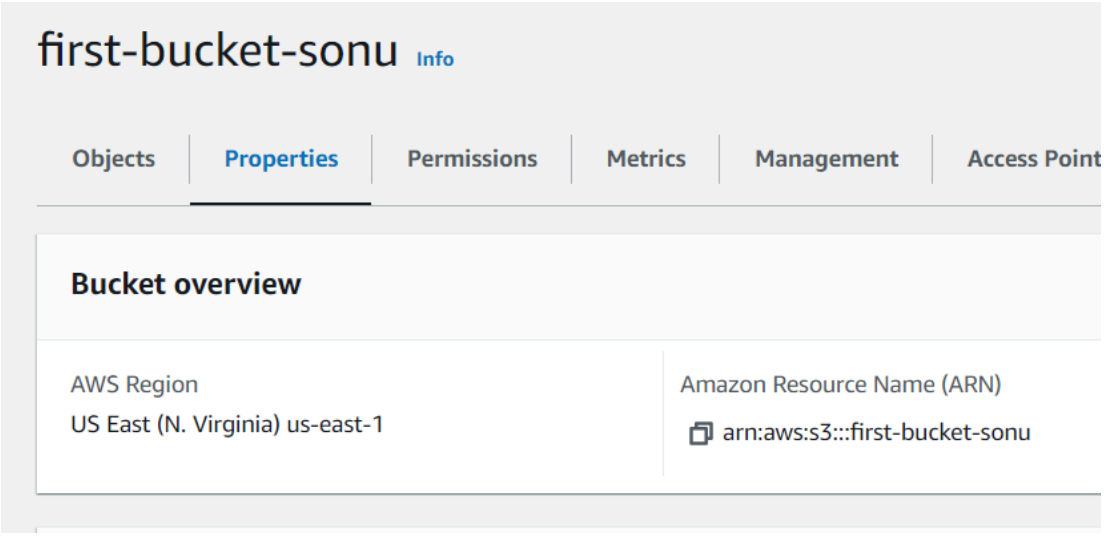
10) Select Policy generator



11) Select policy type as “S3 Bucket Policy”.Fill the form as required. In effect “deny” and in action “PutPbject” is selected.

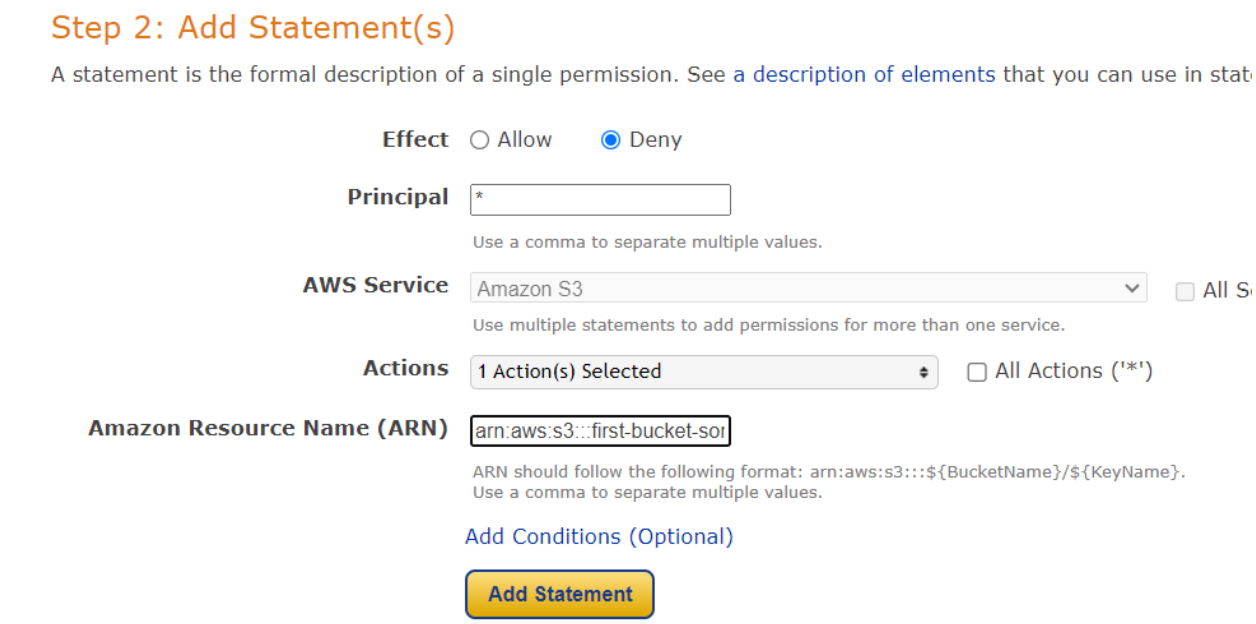


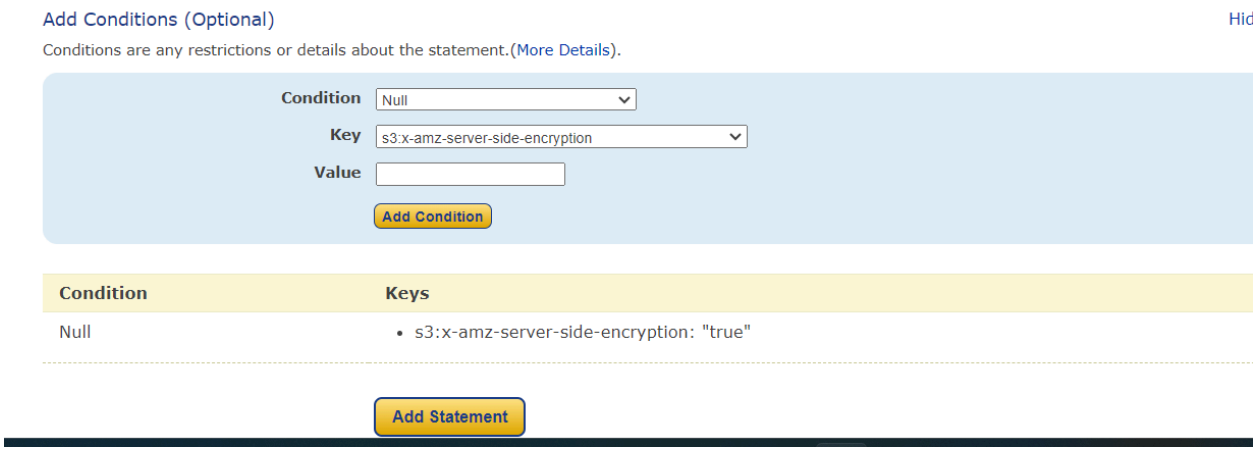
12) Copy the ARN value of s3 bucket from properties section.



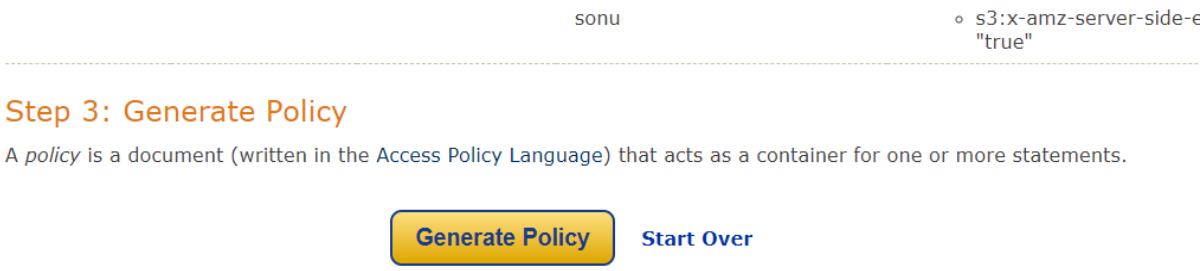
13 Add /* after bucket ARN value.Then, add conditions for denying specific object.

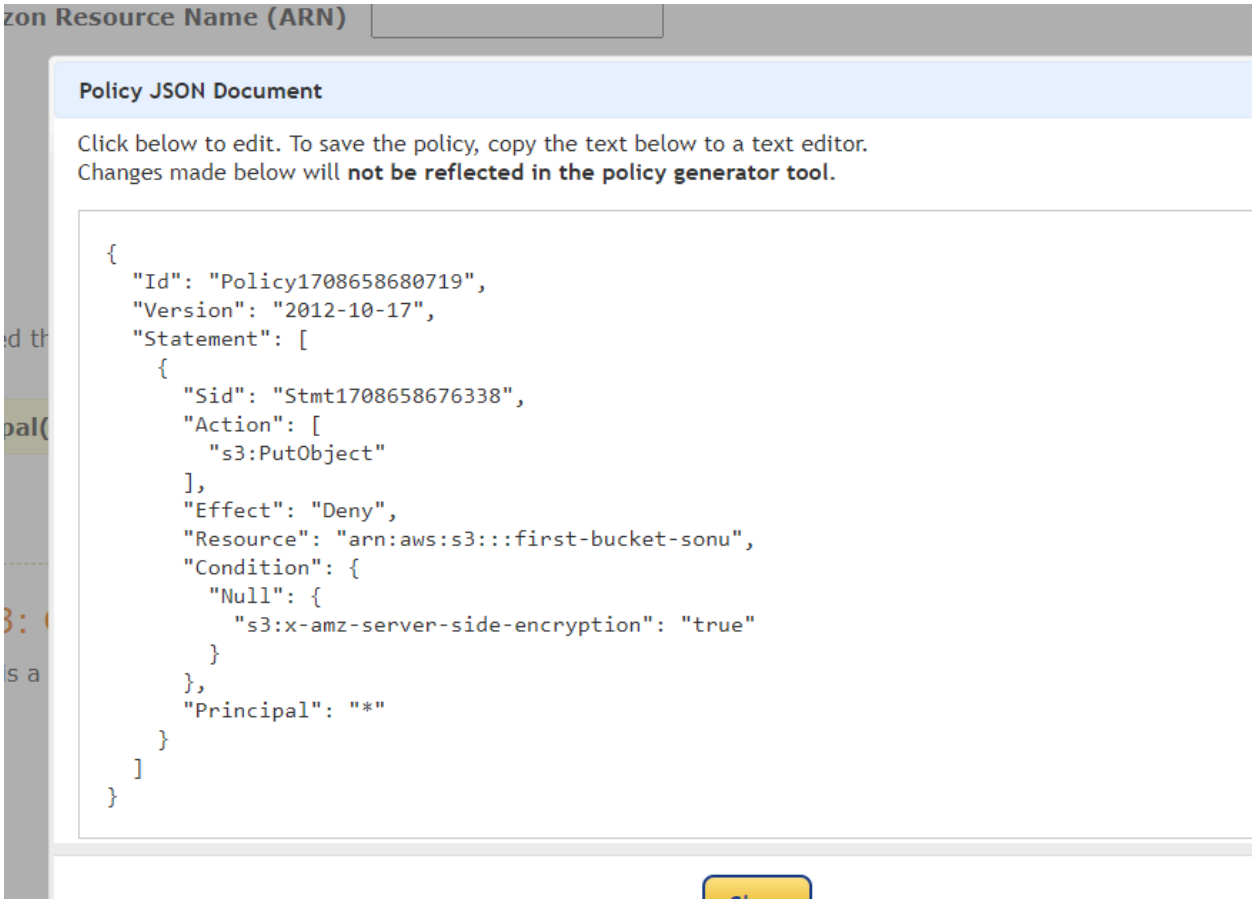
Here, we are denying any objects with Key s3:x amz server side encryption set to Null.





14) Click on Generate Policy and copy the JSON policy document.





15) Add the copied json document into the bucket policy.And click on “Save changes” button.

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bu

Bucket ARN

arn:aws:s3:::first-bucket-sonu

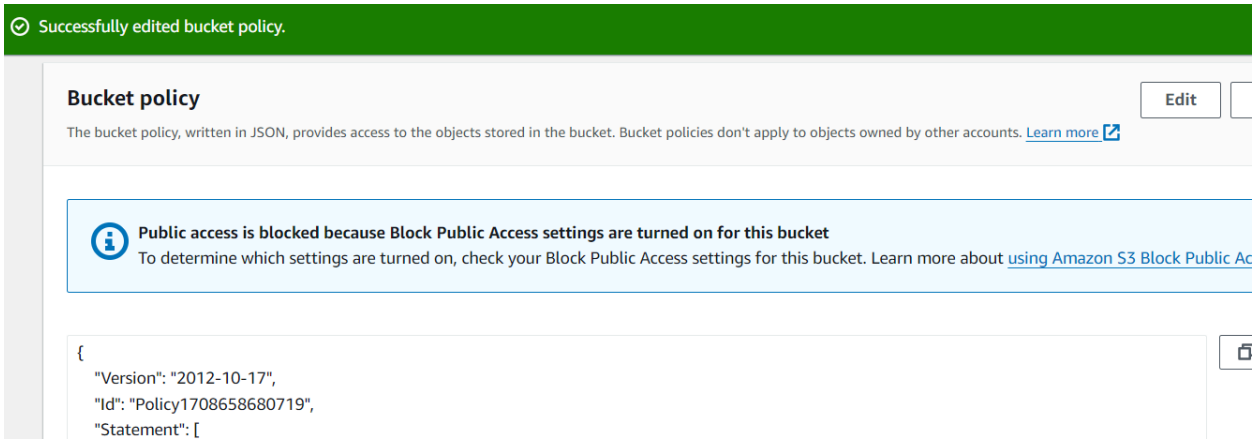
Policy

```
1 {
2   "Id": "Policy1708658680719",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1708658676338",
7       "Action": [
8         "s3:PutObject"
9       ],
```

Preview external access

Cancel Save changes

The bucket policy has been successfully edited.



For testing the policy, we can upload the same html file without specifying any encryption key.The access is denied in this condition.The file can be uploaded successfully only after adding encryption key otherwise it throws an error.