Sneha Reddy

BCSE1944

BTECH CSE (SEM 7)

Lab Assignment 4a

AIM:

A. Create Bucket, upload text file and image file

B. Create version of text file & image file and check for deletion process

C. Enable the public access of the object

D. Explore S3 storage class

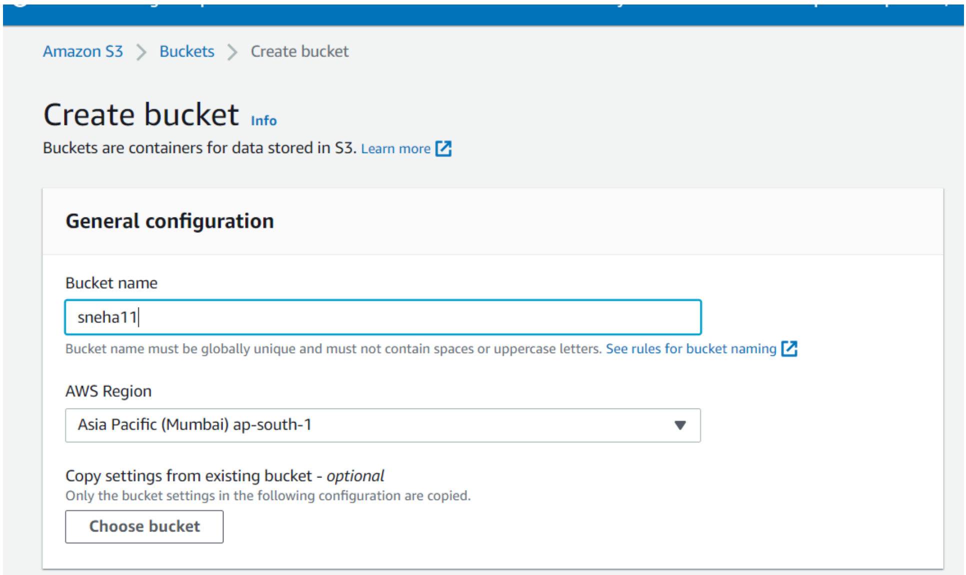
A. Creation of Bucket in AWS

Type S3

Click on Create bucket

There is a rule of creating Bucket's name is that

It should not be Uppercase, no underscore, etc



Buckets are containers for data stored in S3.

General configuration

Bucket name: sneha11

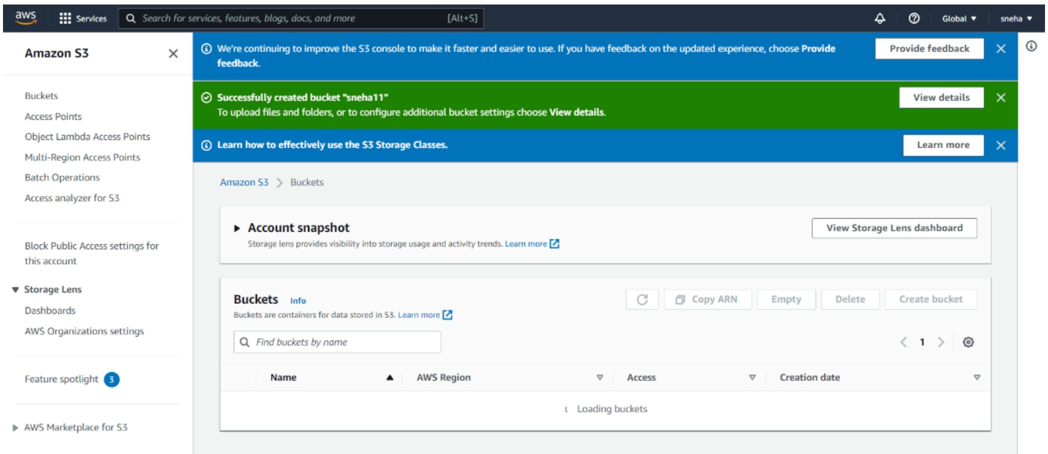
Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Keep all the setting as Default and create a bucket



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully created bucket "sneha11"
To upload files and folders, or to configure additional bucket settings choose **View details**.

Learn how to effectively use the S3 Storage Classes.

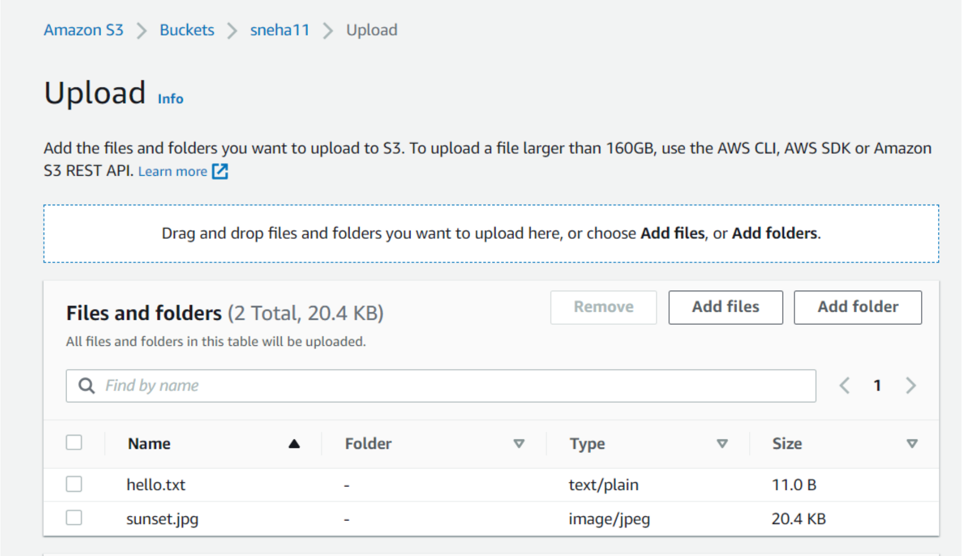
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date

Loading buckets

Upload Text File and click on the bucket that we created and go for the option “Upload Files”. Upload a text file from your machine.



Upload

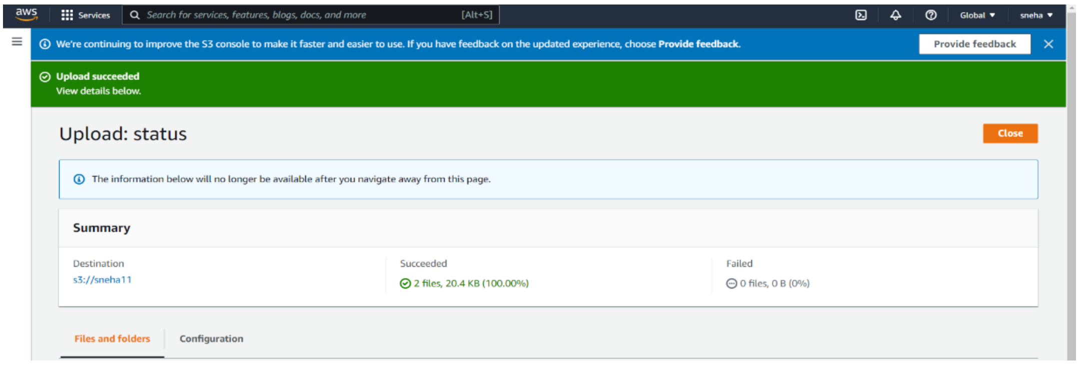
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (2 Total, 20.4 KB)

All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size
□	hello.txt	-	text/plain	11.0 B
□	sunset.jpg	-	image/jpeg	20.4 KB



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Upload succeeded
View details below.

Upload: status

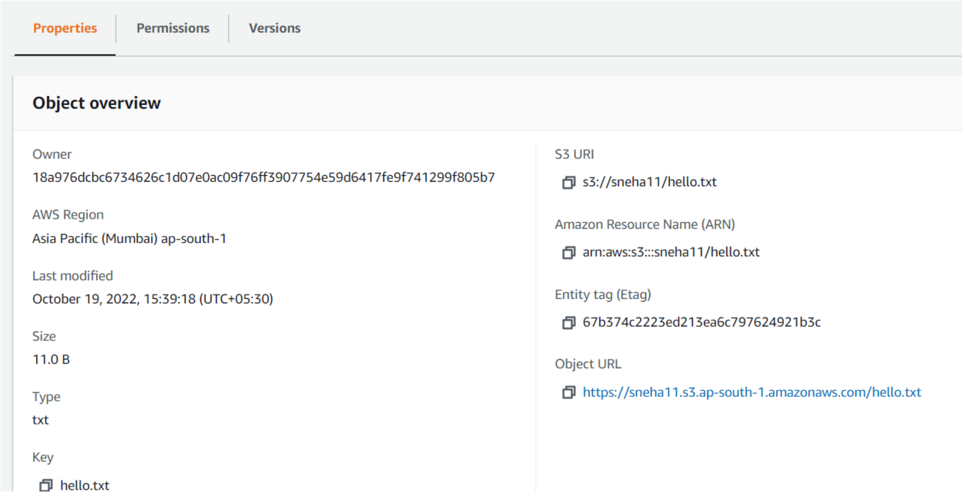
The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://sneha11	2 files, 20.4 KB (100.00%)	0 files, 0 B (0%)

To check whether publicly the bucket is denied or not

Go to Object overview of images or text



Object overview

Owner
18a976dcbc6734626c1d07e0ac09f76ff3907754e59d6417fe9f741299f805b7

AWS Region
Asia Pacific (Mumbai) ap-south-1

Last modified
October 19, 2022, 15:39:18 (UTC+05:30)

Size
11.0 B

Type
txt

Key
hello.txt

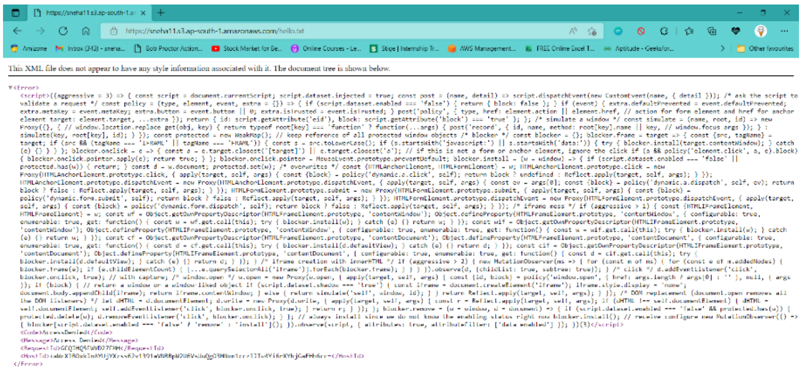
S3 URI
s3://sneha11/hello.txt

Amazon Resource Name (ARN)
arn:aws:s3:::sneha11/hello.txt

Entity tag (Etag)
67b374c2223ed213ea6c797624921b3c

Object URL
https://sneha11.s3.ap-south-1.amazonaws.com/hello.txt

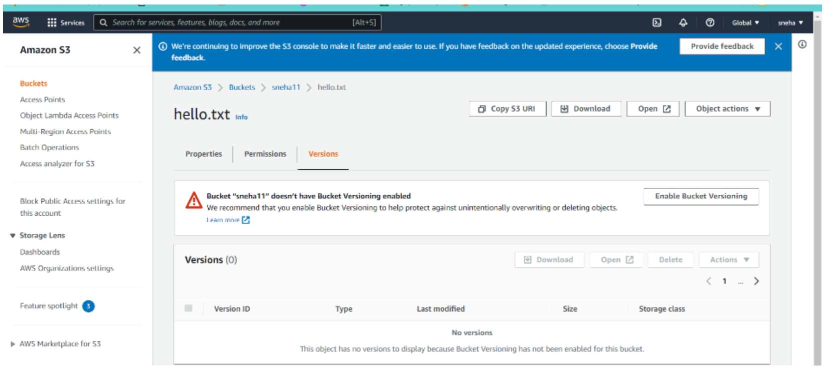
After Clicking on URL – You will get message Access Denied



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

```
▼<Error>
  <script>((aggressive = 3) => { const script = document.currentScript; script.dataset.injected = true; const post = (name, detail) => script.dispatchEvent(new CustomEvent(name, { detail })); /* ask the script to
  validate a request */ const policy = (type, element, event, extra = {}) => { if (script.dataset.enabled === 'false') { return { block: false }; } if (event) { extra.defaultPrevented = event.defaultPrevented;
  extra.metaKey = event.metaKey; extra.button = event.button || 0; extra.isTrusted = event.isTrusted; } post('policy', { type, href: element.action || element.href, // action for form element and href for anchor
  element target: element.target, ...extra }); return { id: script.getAttribute('eid'), block: script.getAttribute('block') === 'true' }; }; /* simulate a window */ const simulate = (name, root, id) => new
  Proxy({}, { // window.location.replace get(obj, key) { return typeof root[key] === 'function' ? function(...args) { post('record', { id, name, method: root[key].name || key, // window.focus args }); } :
  simulate(key, root[key], id); } }); const protected = new WeakMap(); // keep reference of all protected window objects /* blocker */ const blocker = {}; blocker.frame = target => { const {src, tagName} =
  target; if (src && (tagName === 'IFRAME' || tagName === 'FRAME')) { const s = src.toLowerCase(); if (s.startsWith('javascript:') || s.startsWith('data:')) { try { blocker.install(target.contentWindow); } catch
  (e) {} } } }; blocker.onclick = e => { const a = e.target.closest('[target]') || e.target.closest('a'); // if this is not a form or anchor element, ignore the click if (a && policy('element.click', a, e).block)
  { blocker.onclick.pointer.apply(e); return true; } }; blocker.onclick.pointer = MouseEvent.prototype.preventDefault; blocker.install = (w = window) => { if (script.dataset.enabled === 'false' ||
  protected.has(w)) { return; } const d = w.document; protected.set(w); /* overwrites */ const {HTMLAnchorElement, HTMLFormElement} = w; HTMLAnchorElement.prototype.click = new
  Proxy(HTMLAnchorElement.prototype.click, { apply(target, self, args) { const {block} = policy('dynamic.a.click', self); return block ? undefined : Reflect.apply(target, self, args); } });
  HTMLAnchorElement.prototype.dispatchEvent = new Proxy(HTMLAnchorElement.prototype.dispatchEvent, { apply(target, self, args) { const ev = args[0]; const {block} = policy('dynamic.a.dispatch', self, ev); return
  block ? false : Reflect.apply(target, self, args); } }); HTMLFormElement.prototype.submit = new Proxy(HTMLFormElement.prototype.submit, { apply(target, self, args) { const {block} =
  policy('dynamic.form.submit', self); return block ? false : Reflect.apply(target, self, args); } }); HTMLFormElement.prototype.dispatchEvent = new Proxy(HTMLFormElement.prototype.dispatchEvent, { apply(target,
  self, args) { const {block} = policy('dynamic.form.dispatch', self); return block ? false : Reflect.apply(target, self, args); } }); /* iframe mess */ if (aggressive > 1) { const {HTMLIFrameElement,
  HTMLFrameElement} = w; const wf = Object.getOwnPropertyDescriptor(HTMLFrameElement.prototype, 'contentWindow'); Object.defineProperty(HTMLFrameElement.prototype, 'contentWindow', { configurable: true,
  enumerable: true, get: function() { const w = wf.get.call(this); try { blocker.install(w); } catch (e) {} return w; } }); const wif = Object.getOwnPropertyDescriptor(HTMLIFrameElement.prototype,
  'contentWindow'); Object.defineProperty(HTMLIFrameElement.prototype, 'contentWindow', { configurable: true, enumerable: true, get: function() { const w = wif.get.call(this); try { blocker.install(w); } catch
  (e) {} return w; } }); const cf = Object.getOwnPropertyDescriptor(HTMLFrameElement.prototype, 'contentDocument'); Object.defineProperty(HTMLFrameElement.prototype, 'contentDocument', { configurable: true,
  enumerable: true, get: function() { const d = cf.get.call(this); try { blocker.install(d.defaultView); } catch (e) {} return d; } }); const cif = Object.getOwnPropertyDescriptor(HTMLIFrameElement.prototype,
  'contentDocument'); Object.defineProperty(HTMLIFrameElement.prototype, 'contentDocument', { configurable: true, enumerable: true, get: function() { const d = cif.get.call(this); try {
  blocker.install(d.defaultView); } catch (e) {} return d; } }); } /* iframe creation with innerHTML */ if (aggressive > 2) { new MutationObserver(ms => { for (const m of ms) { for (const e of m.addedNodes) {
  blocker.frame(e); if (e.childElementCount) { [...e.querySelectorAll('iframe')].forEach(blocker.frame); } } } }).observe(d, {childList: true, subtree: true}); } /* click */ d.addEventListener('click',
  blocker.onclick, true); // with capture; /* window.open */ w.open = new Proxy(w.open, { apply(target, self, args) { const {id, block} = policy('window.open', { href: args.length ? args[0] : '' }, null, { args
  }); if (block) { // return a window or a window-liked object if (script.dataset.shadow === 'true') { const iframe = document.createElement('iframe'); iframe.style.display = 'none';
  document.body.appendChild(iframe); return iframe.contentWindow; } else { return simulate('self', window, id); } } return Reflect.apply(target, self, args); } }); /* DOM replacement (document.open removes all
  the DOM listeners) */ let dHTML = d.documentElement; d.write = new Proxy(d.write, { apply(target, self, args) { const r = Reflect.apply(target, self, args); if (dHTML !== self.documentElement) { dHTML =
  self.documentElement; self.addEventListener('click', blocker.onclick, true); } return r; } }); }; blocker.remove = (w = window, d = document) => { if (script.dataset.enabled === 'false' && protected.has(w)) {
  protected.delete(w); d.removeEventListener('click', blocker.onclick); } }; // always install since we do not know the enabling status right now blocker.install(); // receive configure new MutationObserver(() =>
  { blocker[script.dataset.enabled === 'false' ? 'remove' : 'install'](); }).observe(script, { attributes: true, attributeFilter: ['data-enabled'] }); })(3)</script>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>GCQ1HQ5FWVD27FMH</RequestId>
  <HostId>taWrX1B0xkInAYUjYXzss62v13Q1aWNRBpW2U6VsUuQgO3MNxn1crz1ITw4Yi6rXYhjGaFHh6rc=</HostId>
</Error>
```

Select the image/ file that you uploaded and then go to the "versions"



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

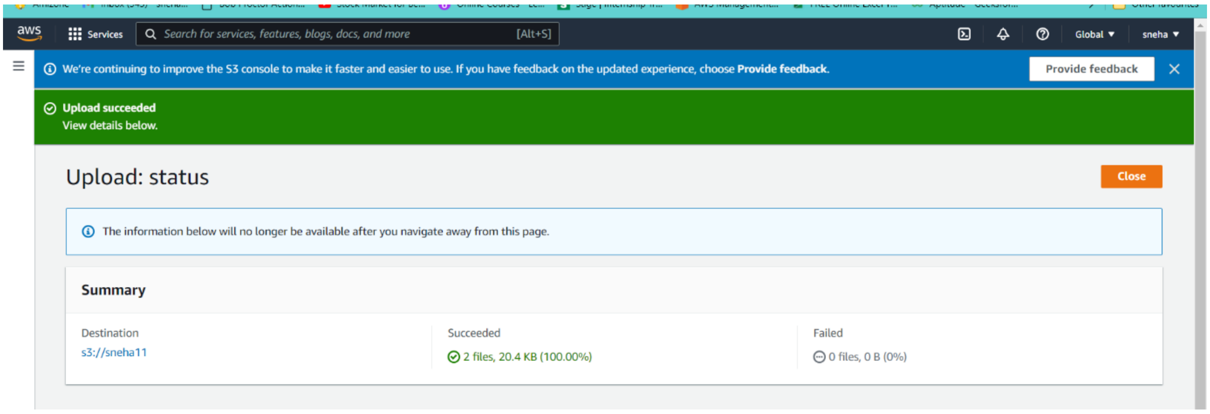
hello.txt Info

Bucket "sneha11" doesn't have Bucket Versioning enabled
We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects.

Versions (0)

□	Version ID	Type	Last modified	Size	Storage class

No versions
This object has no versions to display because Bucket Versioning has not been enabled for this bucket.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
sneha
Provide feedback
Close

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback.**

Upload succeeded
View details below.

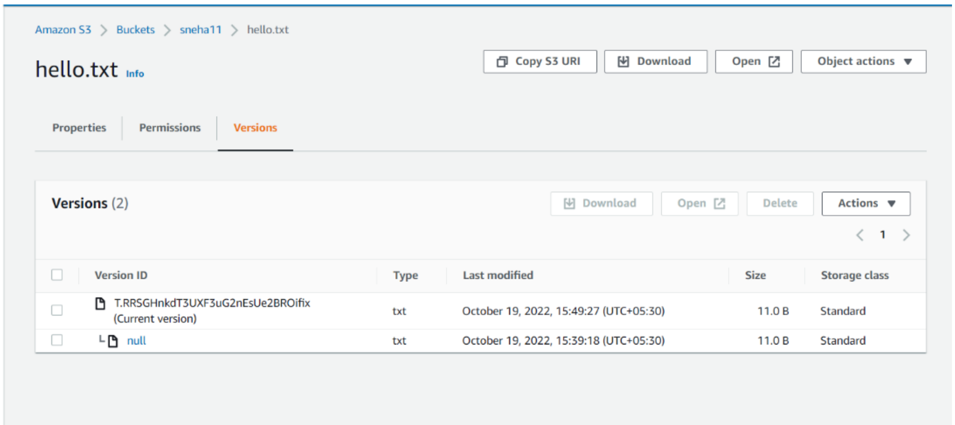
Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://sneha11	2 files, 20.4 KB (100.00%)	0 files, 0 B (0%)

Go to versions of any text/image file to check whether different files have been created or not.



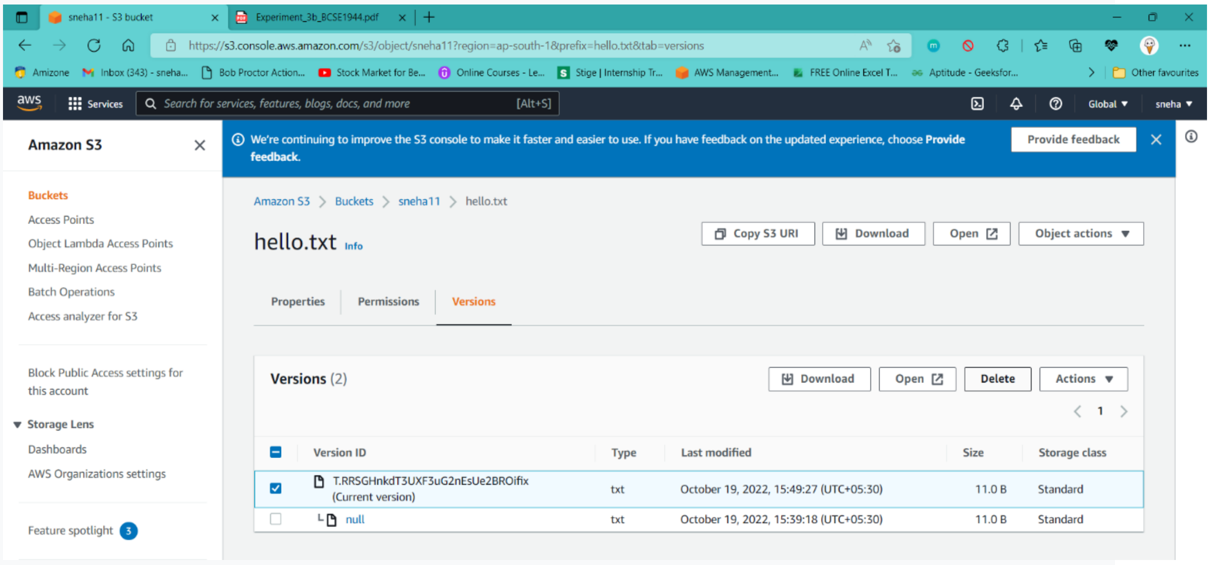
hello.txt

Versions (2)

Version ID	Type	Last modified	Size	Storage class
T.RRSGHnkdT3UXF3uG2nEsUe2BROifix (Current version)	txt	October 19, 2022, 15:49:27 (UTC+05:30)	11.0 B	Standard
null	txt	October 19, 2022, 15:39:18 (UTC+05:30)	11.0 B	Standard

B. Deletion Process

Go to the image/text file and then to the “versions”.

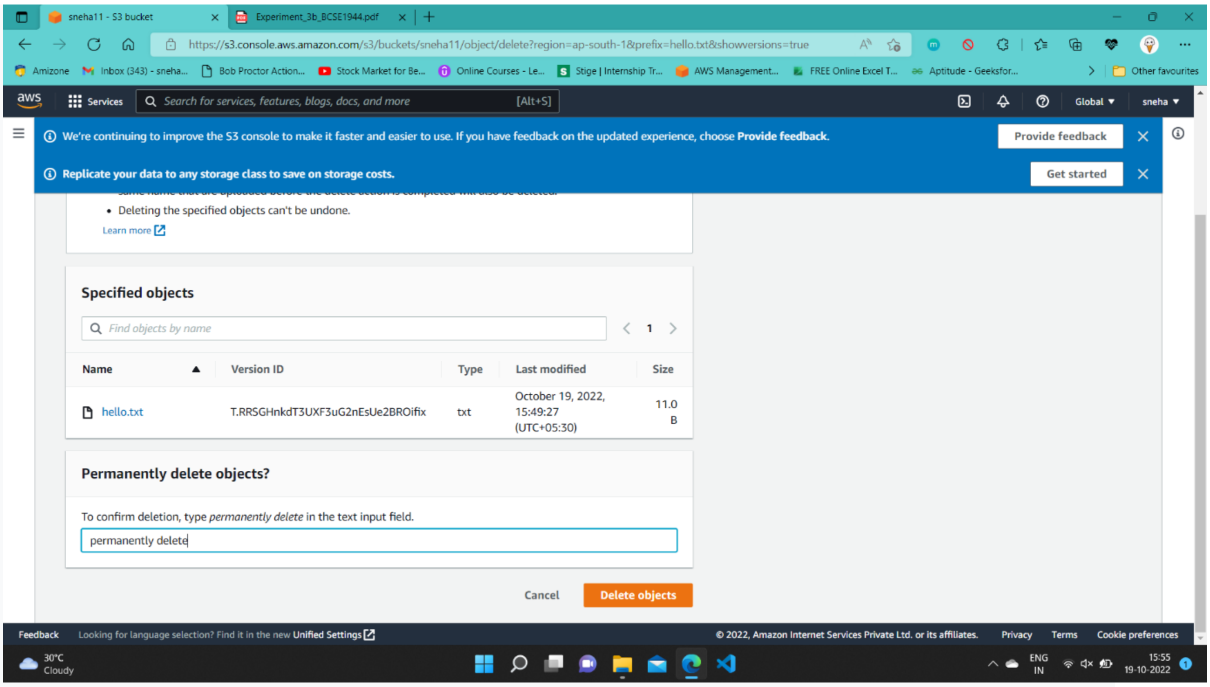


We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback.**

hello.txt

Versions (2)

	Version ID	Type	Last modified	Size	Storage class
☑	T.RRSGHnkdT3UXF3uG2nEsUe2BROifix (Current version)	txt	October 19, 2022, 15:49:27 (UTC+05:30)	11.0 B	Standard
□	null	txt	October 19, 2022, 15:39:18 (UTC+05:30)	11.0 B	Standard



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Replicate your data to any storage class to save on storage costs.

- Deleting the specified objects can't be undone.

Specified objects

Find objects by name

Name ▲	Version ID	Type	Last modified	Size
hello.txt	T.RRSGHnkdT3UXF3uG2nEsUe2BROifix	txt	October 19, 2022, 15:49:27 (UTC+05:30)	11.0 B

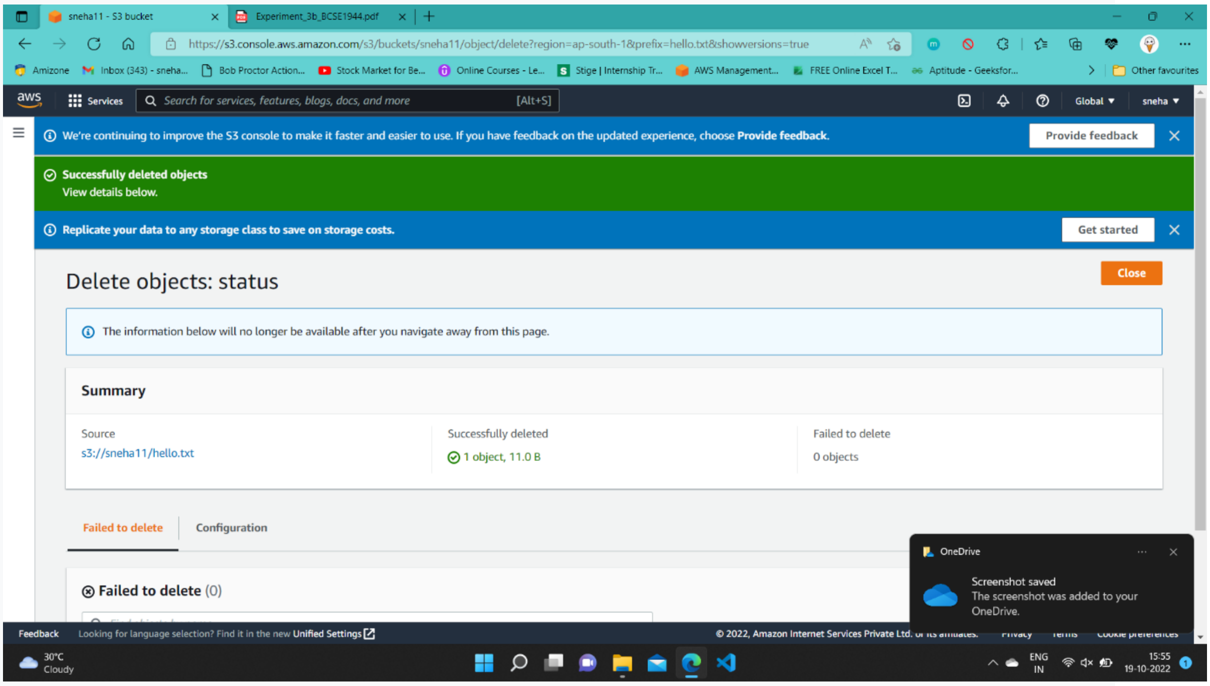
Permanently delete objects?

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Internet Services Private Ltd. or its affiliates.


sneha11 - S3 bucket
Experiment_3b_BCSE1944.pdf
https://s3.console.aws.amazon.com/s3/buckets/sneha11/object/delete?region=ap-south-1&prefix=hello.txt&showversions=true
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▼
sneha ▼
Provide feedback
Get started
Close
Failed to delete
Configuration
OneDrive
Feedback
30°C
Cloudy
ENG
IN
15:55
19-10-2022

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully deleted objects
View details below.

Replicate your data to any storage class to save on storage costs.

Delete objects: status

The information below will no longer be available after you navigate away from this page.

Summary

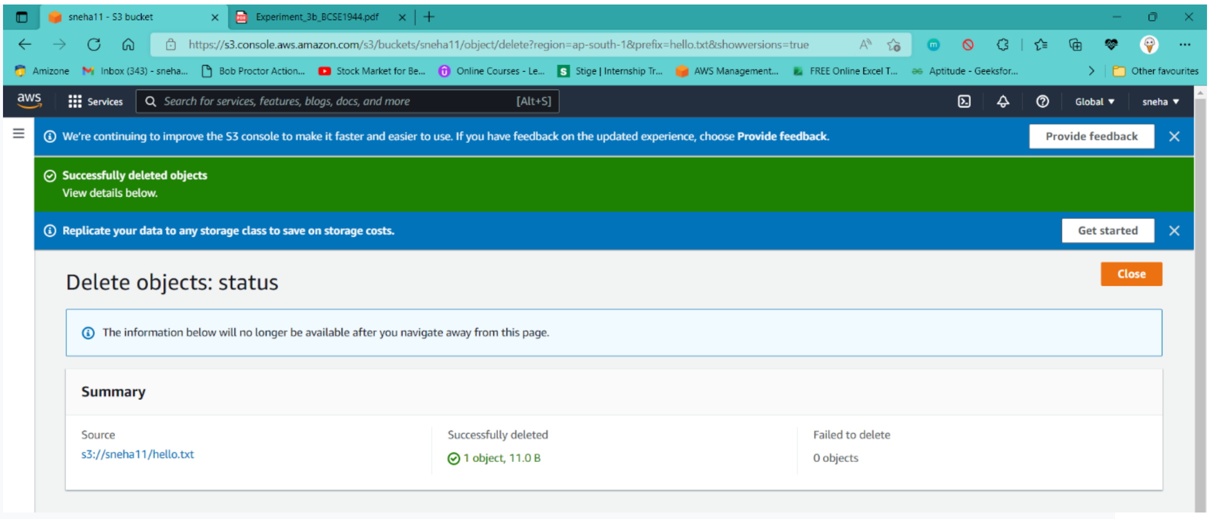
Source	Successfully deleted	Failed to delete
s3://sneha11/hello.txt	✓ 1 object, 11.0 B	0 objects

Failed to delete (0)

Screenshot saved
The screenshot was added to your OneDrive.

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Internet Services Private Ltd.



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully deleted objects
View details below.

Replicate your data to any storage class to save on storage costs.

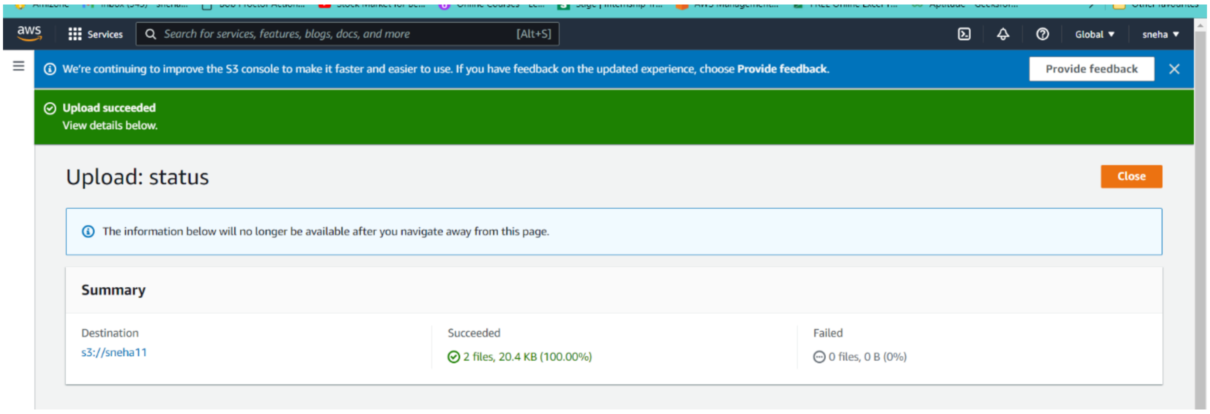
Delete objects: status

The information below will no longer be available after you navigate away from this page.

Summary

Source	Successfully deleted	Failed to delete
s3://sneha11/hello.txt	✓ 1 object, 11.0 B	0 objects

Re-upload the image/text file that you deleted again to check whether the name of the version changes or not



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Upload succeeded
View details below.

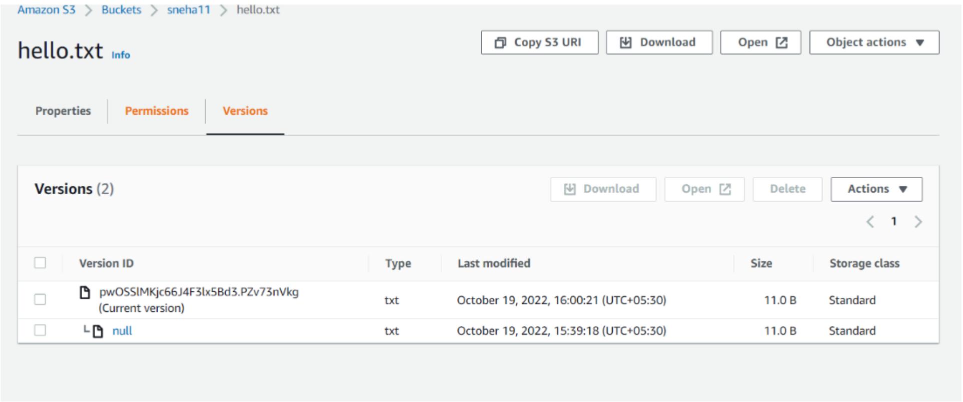
Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://sneha11	✓ 2 files, 20.4 KB (100.00%)	0 files, 0 B (0%)

The version name has been changed.



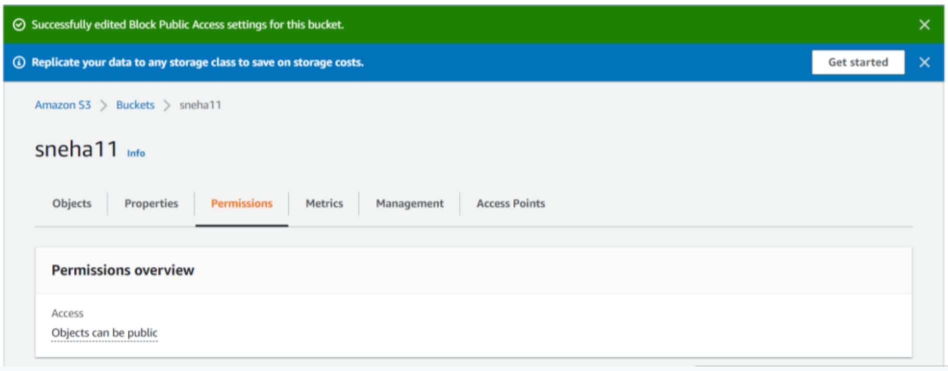
hello.txt

Versions (2)

□	Version ID	Type	Last modified	Size	Storage class
□	pwOSSlMKjc66J4F3lx5Bd3.PZv73nVkg (Current version)	txt	October 19, 2022, 16:00:21 (UTC+05:30)	11.0 B	Standard
□	null	txt	October 19, 2022, 15:39:18 (UTC+05:30)	11.0 B	Standard

C. Enable the public access of the object

Firstly, we need to create bucket policy and for that, go to “permissions” of the bucket you created.



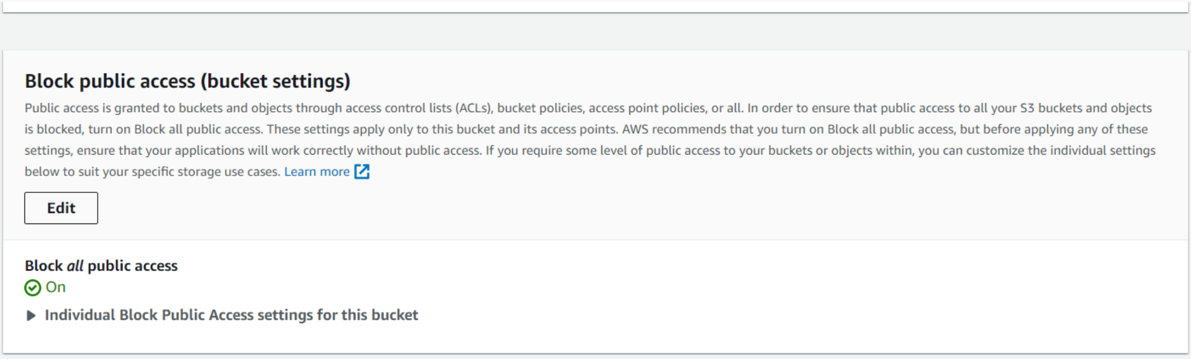
Successfully edited Block Public Access settings for this bucket.

Replicate your data to any storage class to save on storage costs.

Permissions overview

Access
Objects can be public

Go to edit option in “block public access”



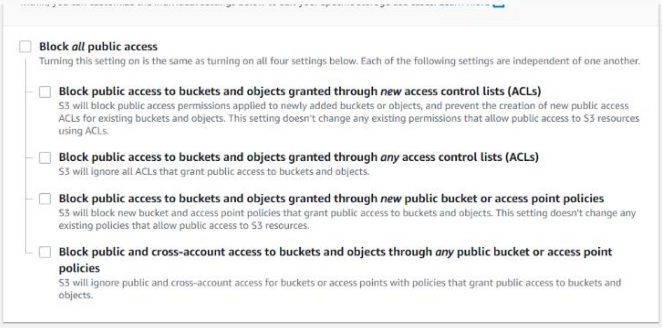
Block public access (bucket settings)

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 *all* public access
On

▶ Individual Block Public Access settings for this bucket

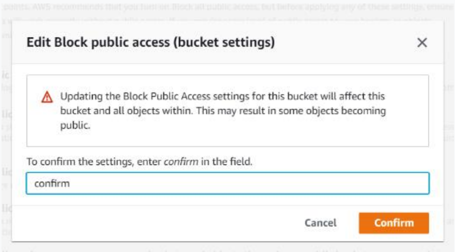
Uncheck the box of “block all public access” and save the changes.



□ 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.

Type “confirm” to allow public access.

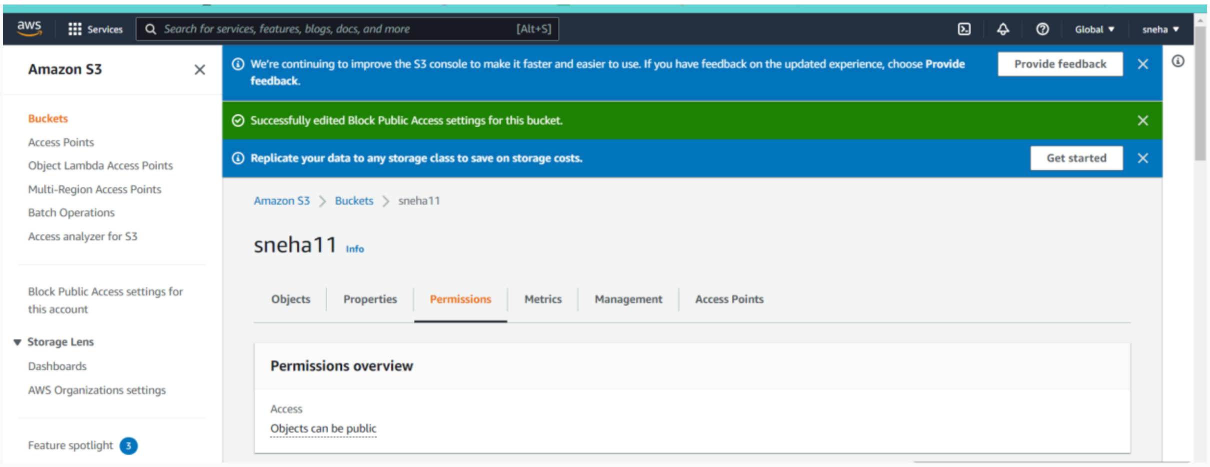


Edit Block public access (bucket settings)

Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public.

To confirm the settings, enter *confirm* in the field.

confirm



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully edited Block Public Access settings for this bucket.

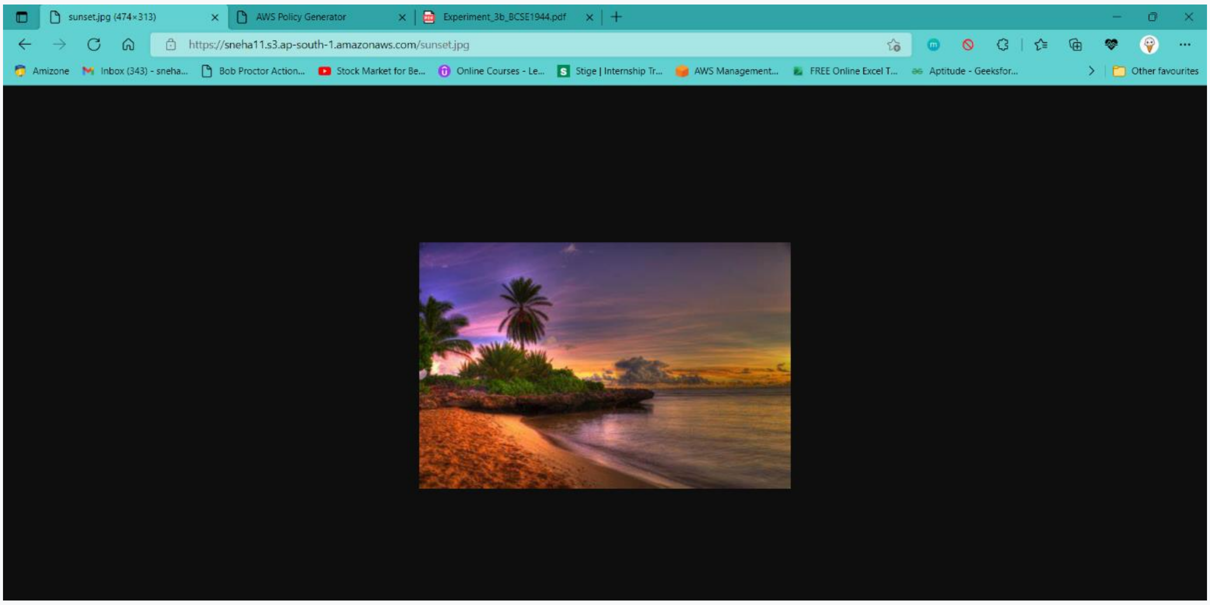
Replicate your data to any storage class to save on storage costs.

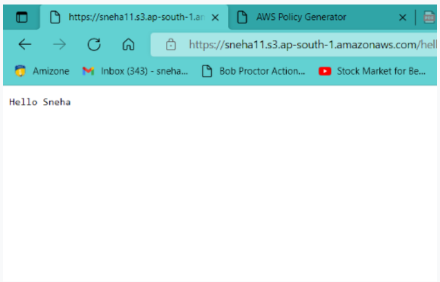
sneha11

Permissions overview

Access
Objects can be public

Now it’s Public Open text file and Image



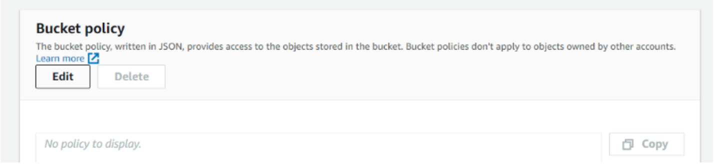


```
Hello Sneha
```

Bucket Policy

Now go to “bucket policy” which is the permissions of the bucket you created

Go to the edit option in the bucket policy

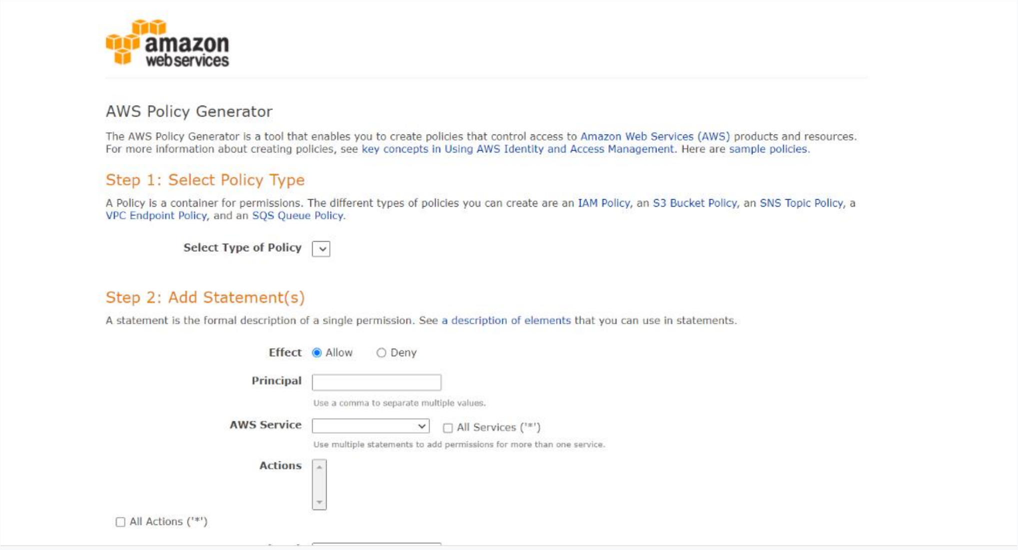


Bucket policy

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

No policy to display.

Copy the URL of “bucket ARN” and go to “policy generator”



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect ● Allow ○ Deny

Principal

Use a comma to separate multiple values.

AWS Service □ All Services ('*')

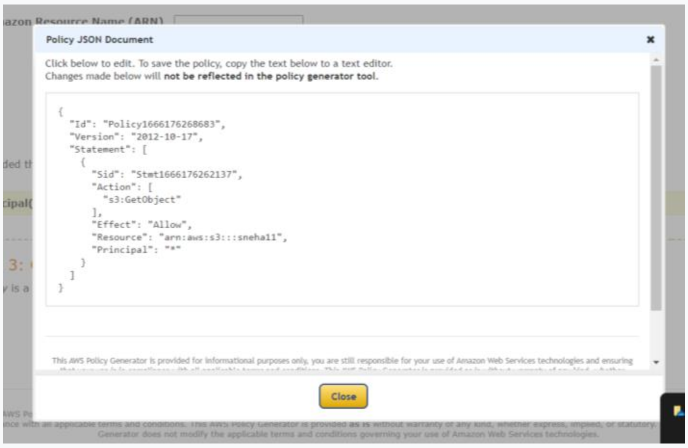
Use multiple statements to add permissions for more than one service.

Actions

□ All Actions ('*')

Now generate policy by selecting policy type and adding statements:

Principals: *
Effect: allow
Actions: Get Object
AWS Service: Amazon S3
ARN: the URL that you copied and add /*



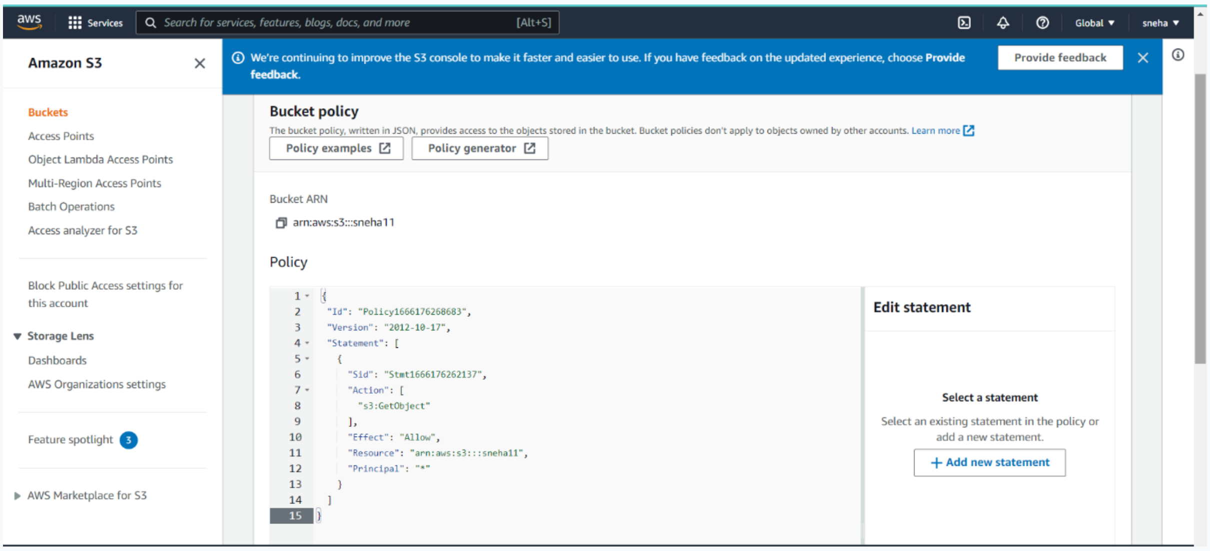
Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id": "Policy1666176268683",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1666176262137",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::sneha11",
      "Principal": "*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring

Copy the policy and past it in bucket policy and save the changes. Finally, the bucket policy will be edited, and your bucket will be accessed publicly.



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

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

Bucket ARN

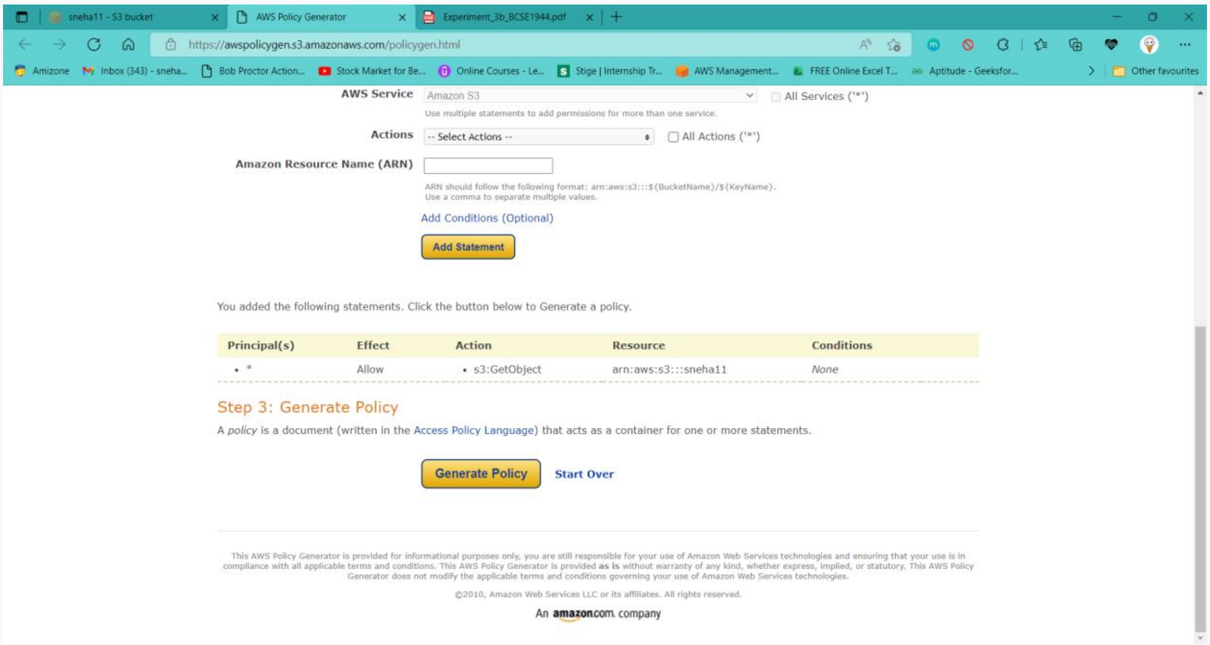
arn:aws:s3:::sneha11

Policy

```
{
  "Id": "Policy1666176268683",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1666176262137",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::sneha11",
      "Principal": "*"
    }
  ]
}
```

Select a statement

Select an existing statement in the policy or add a new statement.



AWS Service: Amazon S3 □ All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions: -- Select Actions -- □ All Actions ('*')

Amazon Resource Name (ARN):

ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.

Add Conditions (Optional)

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetObject	arn:aws:s3:::sneha11	*None*

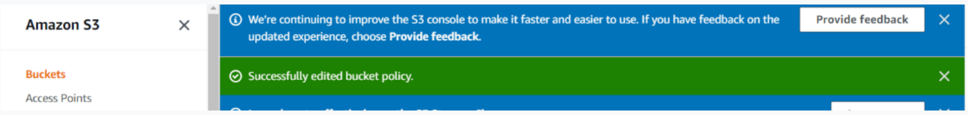
Step 3: Generate Policy

A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided **as is** without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

An amazon.com company



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Successfully edited bucket policy.

D. Exploring S3 storage classes

Amazon Simple Storage Service (S3) is used for storing data in the form of objects S3 is quite different from any other file storage device or service. Amazon S3 also provides industry-leading scalability, data availability, security, and performance. The data, which is uploaded by the user in S3, that data is stored as objects and provided an ID. Moreover, they store in shapes like buckets and can upload the maximum file size is of 5 Terabyte (TB). This service is basically designed for the online backup and archiving of data and applications on Amazon Web Services (AWS).

Amazon S3 storage classes are purpose-built to provide the lowest cost storage for different access patterns. S3 storage classes are ideal for virtually any use case, including those with demanding performance needs, data residency requirements, unknown or changing access patterns, or archival storage.

The S3 storage classes include S3 Intelligent-Tiering for automatic cost savings for data with unknown or changing access patterns, S3 Standard for frequently accessed data, S3 Standard-Infrequent Access (S3 Standard-IA) and S3 One Zone-Infrequent Access (S3 One Zone-IA) for less frequently accessed data, S3 Glacier Instant Retrieval for archive data that needs immediate access, S3 Glacier Flexible Retrieval (formerly S3 Glacier) for rarely accessed long-term data that does not require immediate access, and Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive) for long-term archive and digital preservation with retrieval in hours at the lowest cost storage in the cloud. If you have data residency requirements that can’t be met by an existing AWS Region, you can use the S3 Outposts storage class to store your S3 data on premises. Amazon S3 also offers capabilities to manage your data throughout its lifecycle. Once an S3 Lifecycle policy is set, your data will automatically transfer to a different storage class without any changes to your application