

## Experiment no 4

Aim :To study and Implement Storage as a Service using Own Cloud/ AWS S3, Glaciers/ Azure Storage.

### Theory:

#### **Storage as a service (STaaS)**

Storage as a service (STaaS) is a cloud business model in which a company leases or rents its storage infrastructure to another company or individuals to store data.

Small companies and individuals often find this to be a convenient methodology for managing backups, and providing cost savings in personnel, hardware and physical space.

As an alternative to storing magnetic tapes offsite in a vault, IT administrators are meeting their storage and backup needs by service level agreements (SLAs) with an SaaS provider, usually on a cost-per-gigabyte-stored and cost-per-data-transferred basis. The client transfers the data meant for storage to the service provider on a set schedule over the SaaS provider's wide area network or over the Internet.

#### *Advantages of STaaS*

1. Storage costs  
Personnel, hardware and physical storage space expenses are reduced.
2. Disaster recovery  
Having multiple copies of data stored in different locations can better enable disaster recovery measures.
3. Scalability  
With most public cloud services, users only pay for the resources that they use.
4. Syncing  
Files can be automatically synced across multiple devices.
5. Security  
Security can be both an advantage and a disadvantage, as security methods may change per vendor. Data tends to be encrypted during transmission and while at rest.

#### *Disadvantages of STaaS*

1. Security  
Users may end up transferring business-sensitive or mission-critical data to the cloud, which makes it important to choose a service provider that's reliable.
2. Potential storage costs  
If bandwidth limitations are exceeded, these could be expensive.
3. Potential downtimes  
Vendors may go through periods of downtime where the service is not available, which can be trouble for mission-critical data.
4. Limited customization  
Since the cloud infrastructure is owned and managed by the service provider, it is less

customizable.

5. Potential for vendor lock-in

It may be difficult to migrate from one service to another.

## **Amazon S3**

Amazon S3 (Simple Storage Service) provides object storage, which is built for storing and recovering any amount of information or data from anywhere over the internet. It provides this storage through a web services interface. While designed for developers for easier web-scale computing, it provides 99.999999999 percent durability and 99.99 percent availability of objects. It can also store computer files up to 5 terabytes in size.

### **AWS S3 Benefits**

- Durability: S3 provides 99.999999999 percent durability.
- Low cost: S3 lets you store data in a range of “storage classes.” These classes are based on the frequency and immediacy you require in accessing files.
- Scalability: S3 charges you only for what resources you actually use, and there are no hidden fees or overage charges. You can scale your storage resources to easily meet your organization’s ever-changing demands.
- Availability: S3 offers 99.99 percent availability of objects
- Security: S3 offers an impressive range of access management tools and encryption features that provide top-notch security.
- Flexibility: S3 is ideal for a wide range of uses like data storage, data backup, software delivery, data archiving, disaster recovery, website hosting, mobile applications, IoT devices, and much more.
- Simple data transfer: You don’t have to be an IT genius to execute data transfers on S3. The service revolves around simplicity and ease of use.

## Implementation:

### Step 1: In AWS, Services->Storage-> S3

The screenshot shows the AWS S3 console interface. On the left, there is a sidebar with the following sections and links:

- Buckets**
  - Access Points
  - Object Lambda Access Points
  - Multi-Region Access Points
  - Batch Operations
  - Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens**
  - Dashboards
  - AWS Organizations settings
- Feature spotlight
- AWS Marketplace for S3

At the bottom of the sidebar, there are links for Feedback, English (US), and cookie preferences.

The main content area is titled "Amazon S3" and "Amazon S3". It features an "Account snapshot" section with a link to "View Storage Lens dashboard". Below this is a "Buckets (0) Info" section with a note that buckets are containers for data stored in S3. It includes a search bar, a "Create bucket" button, and a table header for "Name", "AWS Region", "Access", and "Creation date". A message states "No buckets" and "You don't have any buckets." with another "Create bucket" button.

The footer of the page includes copyright information for 2022, links for Privacy, Terms, and Cookie preferences, and a note about mobile devices.

### Step 2: Click on Create bucket

### Step 3: Adding Bucket name and choosing AWS Region

AWS Services Search for services, features, blogs, docs, and more [Alt+S] Global ▾ Achu S J ▾

Amazon S3 > Create bucket

## Create bucket Info

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

### General configuration

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

AWS Region

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership

Feedback English (US) ▾ © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

## Step 4: Enable Bucket Versioning

The screenshot shows the AWS S3 Bucket Properties page for a bucket named 'aws-tutorial'. The 'Bucket Versioning' section is highlighted, showing the 'Enable' option selected. Below it, the 'Tags (0) - optional' section indicates no tags are associated with the bucket. The 'Default encryption' section shows that objects are automatically encrypted. The bottom navigation bar includes links for Feedback, English (US), Privacy, Terms, and Cookie preferences.

**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**  
 Disable  
 Enable

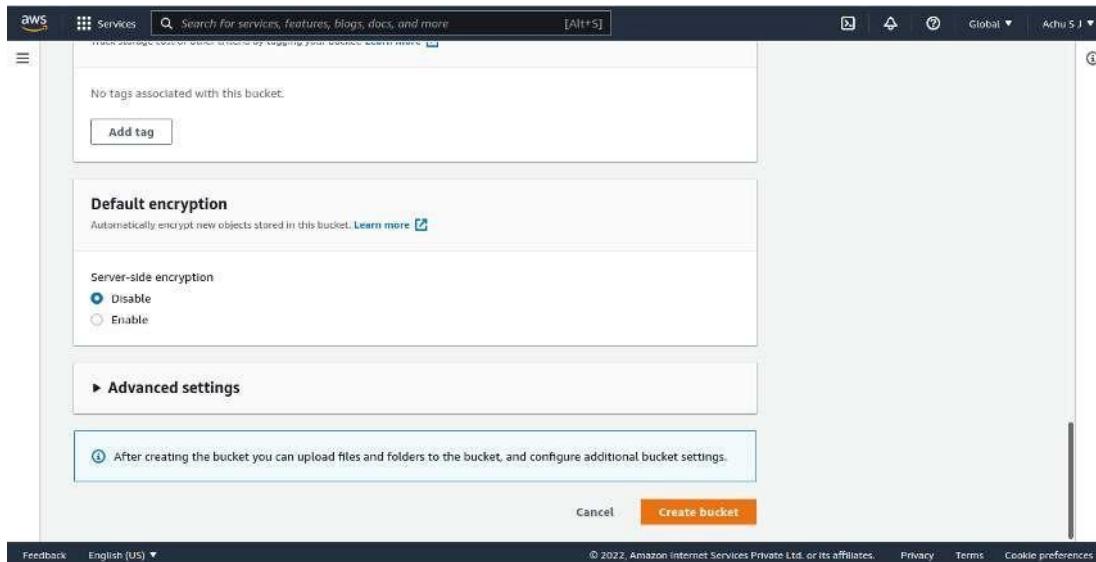
**Tags (0) - optional**  
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.  
[Add tag](#)

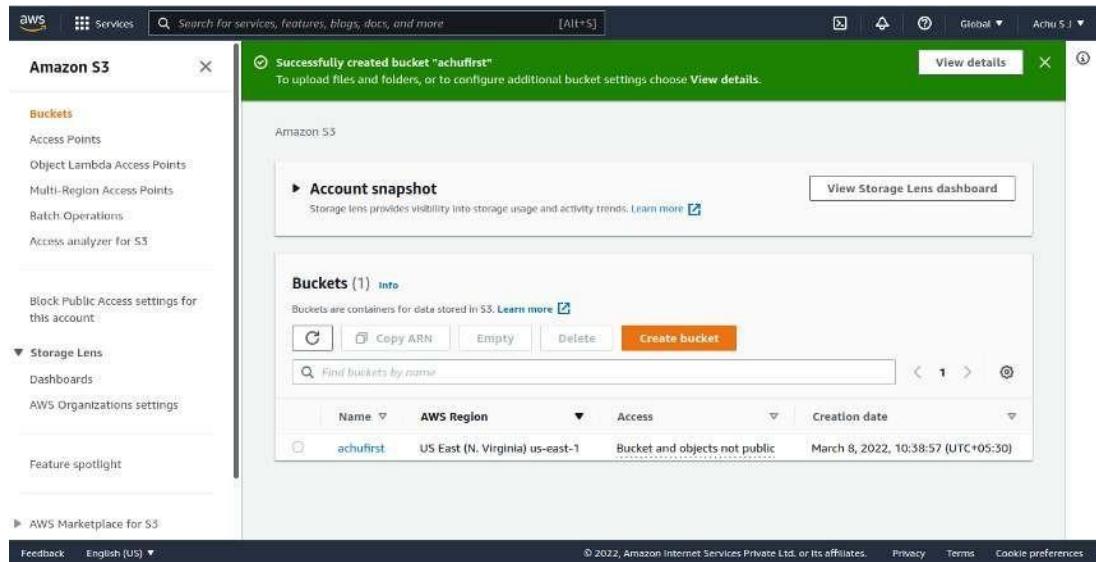
**Default encryption**  
Automatically encrypt new objects stored in this bucket. [Learn more](#)

Feedback English (US) © 2022, Amazon Internet Services Private Ltd, or its affiliates. Privacy Terms Cookie preferences

## Step 5: Disable Default encryption and click Create bucket



## Step 6: Bucket 'achufirst' is created



## Step 7: Selecting 'achufirst' and uploading files

The screenshot shows the AWS S3 console interface. At the top, there's a search bar and navigation links for services and global settings. Below that, the 'Amazon S3' section shows the 'achufirst' bucket. The 'Objects' tab is active, displaying a message stating 'Objects (0)'. It explains that objects are fundamental entities stored in Amazon S3 and provides a link to 'Amazon S3 Inventory'. Below this, there are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar for 'Find objects by prefix' is present, along with a 'Show versions' link. A table header for 'No objects' includes columns for Name, Type, Last modified, Size, and Storage class. A message at the bottom says 'You don't have any objects in this bucket.' An orange 'Upload' button is located at the bottom right of the main content area.

## Step 8: Uploaded files successfully

The screenshot shows the AWS S3 console after a file upload. A green banner at the top says 'Upload succeeded' with a 'View details below.' link. Below it, the 'Upload: status' page has a summary table. The 'Summary' section shows the destination as 's3://achufirst' and lists 'Succeeded' (3 files, 420.3 KB (100.00%)) and 'Failed' (0 files, 0 B (0%)). There are tabs for 'Files and folders' (selected) and 'Configuration'. The 'Files and folders' section shows a table with 3 total files and a total size of 420.3 KB. A search bar for 'Find by name' is at the top of this section. At the bottom, there are links for Feedback, English (US), and various legal and preference links.

**Step 9: After uploading the same png file one by one, clicking on 'Show versions', we can see that the Version ID is different for both**

| Name                                    | Type   | Version ID                       | Last modified                       | Size     | Storage class |
|---|--------|----------------------------------|-------------------------------------|----------|---------------|
| achu-cc/                                | Folder | -                                | -                                   | -        | -             |
| Screenshot from 2022-03-08 10-46-21.png | png    | r9SilmHsbQq4aGK0kVoX6c2XdvD8ZxG4 | March 8, 2022, 10:50:49 (UTC+05:30) | 210.1 KB | Standard      |
| Screenshot from 2022-03-08 10-46-21.png | png    | veTxagG988Ep1k002w12ddqX2q0Fw2HU | March 8, 2022, 10:48:53 (UTC+05:30) | 210.1 KB | Standard      |

**Step 10: Copying ARN for 'achufirst'**

| Name      | AWS Region                      | Access                        | Creation date                       |
|-----------|---------------------------------|-------------------------------|-------------------------------------|
| achufirst | US East (N. Virginia) us-east-1 | Bucket and objects not public | March 8, 2022, 10:38:57 (UTC+05:30) |

**Step 11: Going to Permissions -> Edit Bucket Policy -> Policy Generator Do as shown**



## 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 (\*)

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)

Principal  Use a comma to separate multiple values.

AWS Service   All Services (\*)

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:DeleteObject | arn:aws:s3:::achufirst/* | None       |

### Step 3: Generate Policy

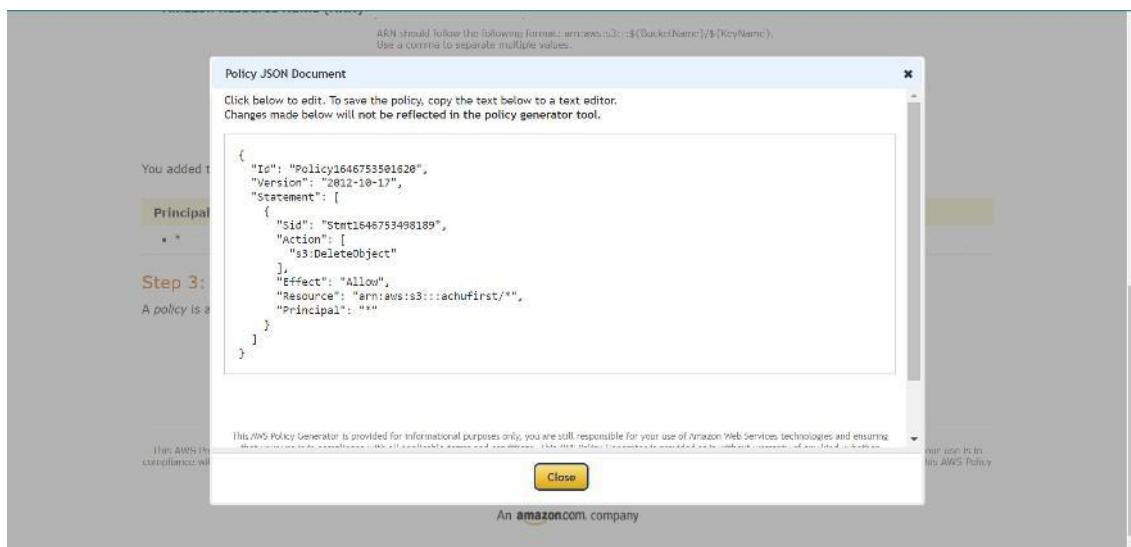
A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

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

**Step 12: Click on Generate Policy and the following JSON document is obtained. It is then copied.**



## Step 13: The bucket policy is updated and changes are saved.

```

1▼ [
2  "Id": "Policy1646753501620",
3  "Version": "2012-10-17",
4  "Statement": [
5    {
6      "Sid": "Stmt1646753498189",
7      "Action": [
8        "s3:DeleteObject"
9      ],
10     "Effect": "Allow",
11     "Resource": "arn:aws:s3:::achufirst/*",
12     "Principal": "*"
13   }
14 ]

```

## Step 14: Successfully edited bucket policy

The screenshot shows the AWS S3 Bucket Policy editor. At the top, a green banner indicates "Successfully edited bucket policy." Below it, a message states: "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." There are "Edit" and "Delete" buttons. A warning box says: "Public access is blocked because Block Public Access settings are turned on for this bucket. To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about using Amazon S3 Block Public Access." The JSON code for the policy is displayed:

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1646717259808",  
    "Statement": [  
        {  
            "Sid": "Stmt1646717201966",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:DeleteObject",  
            "Resource": "arn:aws:s3:::achufirst/*"  
        }  
    ]  
}
```

At the bottom, there are links for "Feedback", "English (US)", and "Cookie preferences".

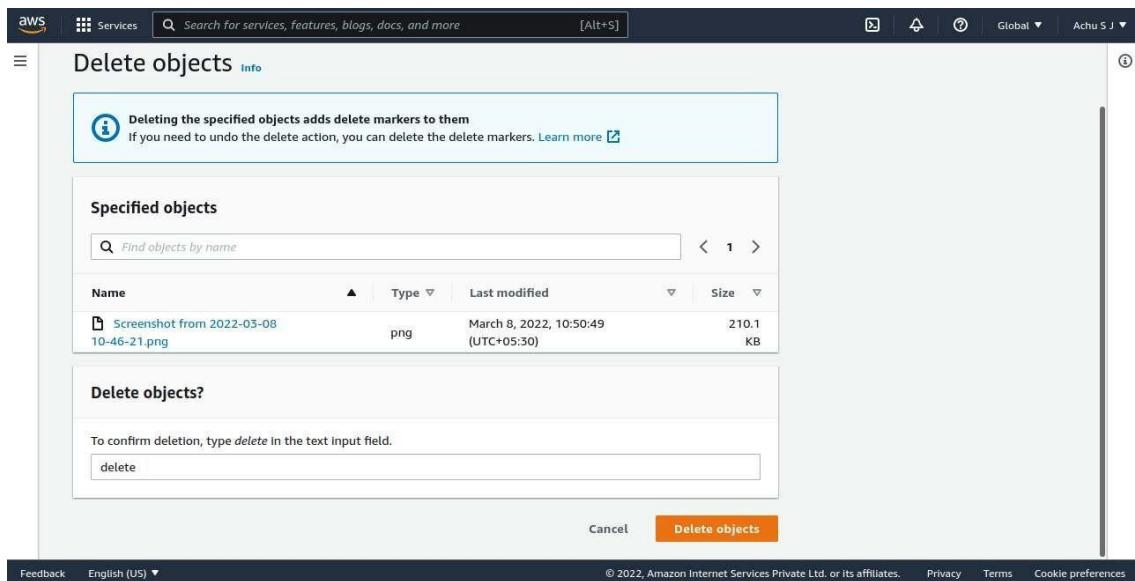
## Step 15: Delete objects inside bucket

The screenshot shows the AWS S3 Buckets page. On the left, a sidebar lists "Buckets", "Storage Lens", and "Feature spotlight". The main area displays an "Account snapshot" with a "View Storage Lens dashboard" button. Below it is a "Buckets (1) Info" section. A message says: "Buckets are containers for data stored in S3. Learn more." It includes buttons for "Create bucket", "Copy ARN", "Empty", and "Delete". A search bar "Find buckets by name" is present. A table lists the single bucket:

| Name      | AWS Region                      | Access                        | Creation date                       |
|-----------|---------------------------------|-------------------------------|-------------------------------------|
| achufirst | US East (N. Virginia) us-east-1 | Bucket and objects not public | March 8, 2022, 10:38:57 (UTC+05:30) |

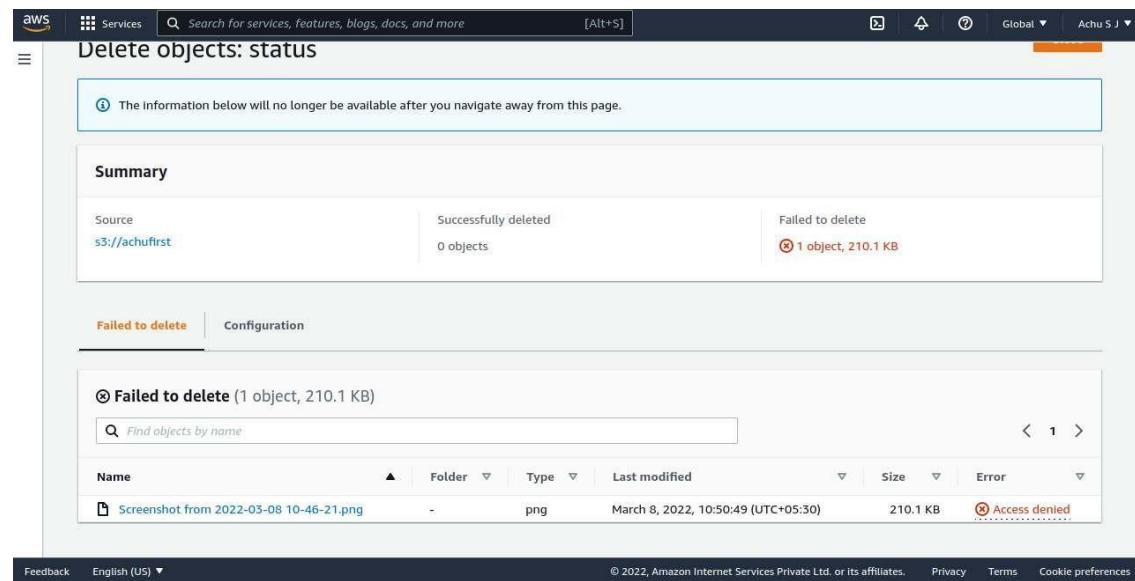
At the bottom, there are links for "Feedback", "English (US)", and "Cookie preferences".

## Step 16: Reviewing objects in bucket and confirming deletion



The screenshot shows the AWS S3 'Delete objects' dialog. At the top, there's a message: 'Deleting the specified objects adds delete markers to them. If you need to undo the delete action, you can delete the delete markers.' Below this is a table titled 'Specified objects' with one item: 'Screenshot from 2022-03-08 10-46-21.png' (png, March 8, 2022, 10:50:49 (UTC+05:30), 210.1 KB). The 'Delete objects?' section contains a text input field with 'delete' typed in. The 'Delete objects' button is at the bottom right.

## Step 17: Deletion is not possible because of the policy we added



The screenshot shows the AWS S3 'Delete objects: status' page. It displays a summary: 'Source s3://achufirst' with 'Successfully deleted 0 objects' and 'Failed to delete 1 object, 210.1 KB'. The 'Failed to delete' tab is selected, showing a table with one row: 'Screenshot from 2022-03-08 10-46-21.png' (png, March 8, 2022, 10:50:49 (UTC+05:30), 210.1 KB, Error: Access denied).

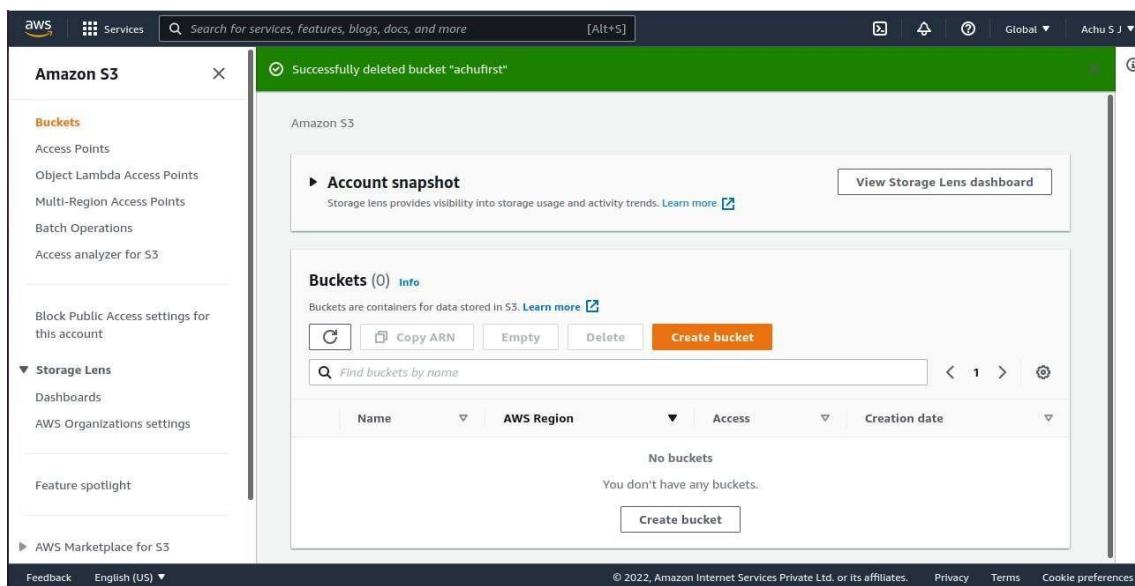
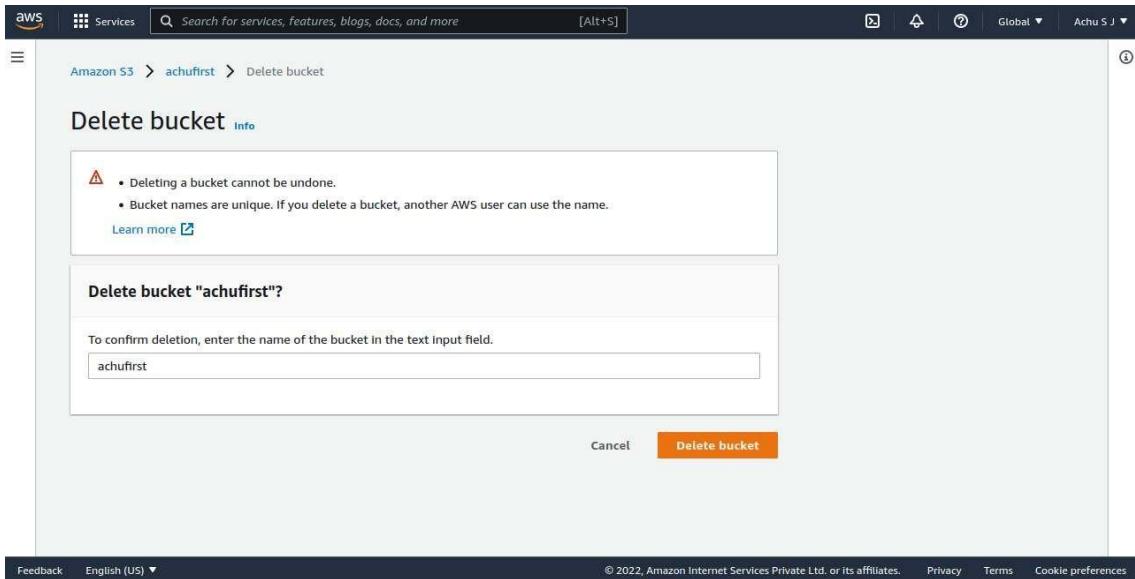
## Step 18: Now, emptying bucket

The screenshot shows the 'Empty bucket' confirmation dialog in the AWS S3 console. At the top, there's a warning message about the permanence of object deletion. Below it, a note suggests using a lifecycle rule for efficiency. The main question asks if you want to permanently delete all objects in the bucket. A text input field contains the text 'permanently delete'. At the bottom right are 'Cancel' and 'Empty' buttons.

**Emptied bucket successfully**

The screenshot shows the 'Empty bucket: status' page after the deletion. It displays a summary table with the source bucket ('s3://achufirst') and the outcome ('Successfully deleted' with 4 objects). Below this, a table titled 'Failed to delete (0)' shows no failed object deletions. The page includes standard navigation and footer links.

## Step 19: Deleting bucket



## Conclusion:

Thus, we have learnt about Storage as a Service and implemented the same using AWS S3.