

## AWS Documentation

No:-	Content
1.	<b>S3(simple storage service)</b>
2.	<b>Create simple bucket</b> <ul style="list-style-type: none"><li>• Publicly available</li><li>• Privately available</li></ul>
3.	<b>Storing logs into another bucket.</b>
4.	<b>Versioning of bucket</b>
5.	<b>Hosting one static website.</b>
6.	<b>Storage classes</b>
7.	<b>Replication of bucket objet</b>

### **S3(simple storage service):**

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

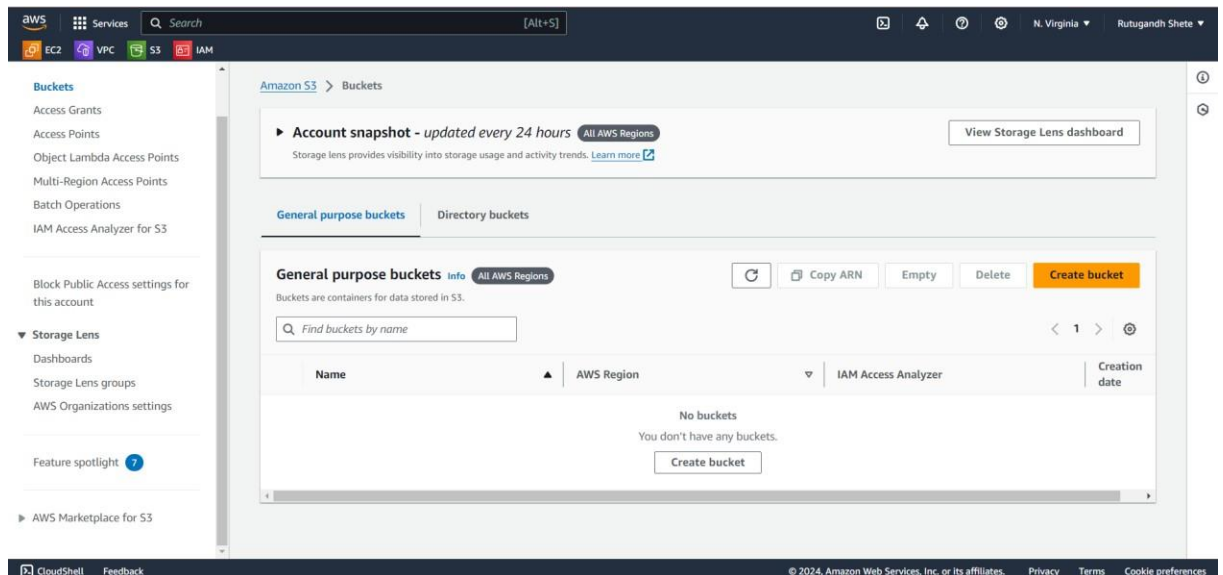


## Create simple bucket:

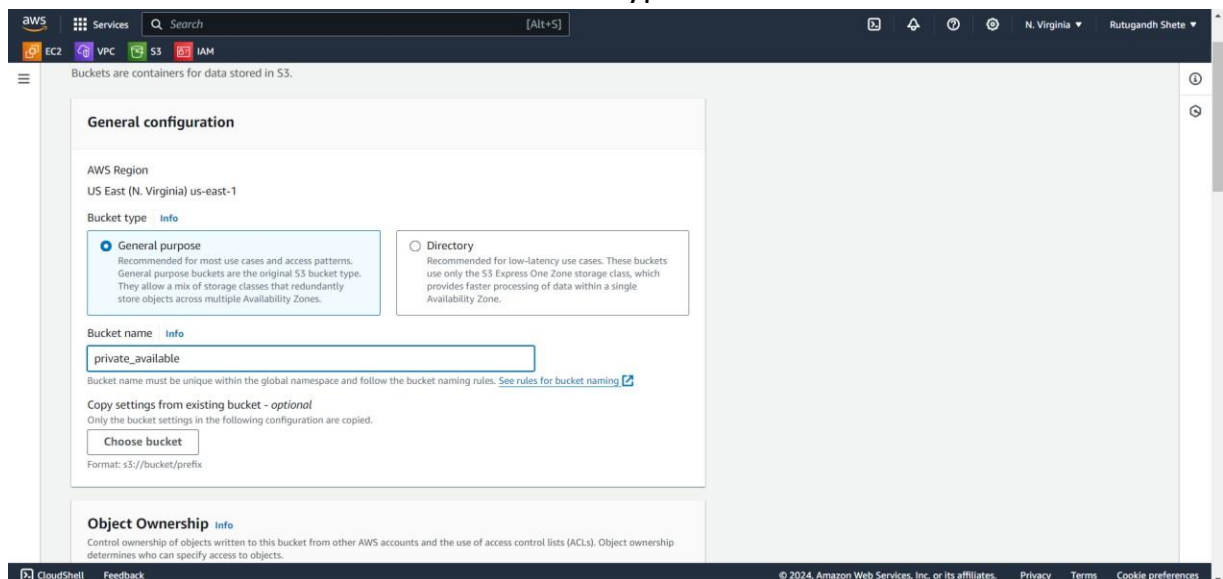
## Private-Bucket

### Steps:

- Go to S3 service.



- Click on create bucket → Name → Bucket type →



- To make bucket private then mention bucket ownership → 1. ACL's is disabled (for private bucket) 2. ACL's is enabled (for public bucket).

### 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 determines who can specify access to objects.

☒ **ACLs disabled (recommended)**  
 All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**  
 Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership  
 Bucket owner enforced

- Check this block to make to private or else we can uncheck it to make it public.

### Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ **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.

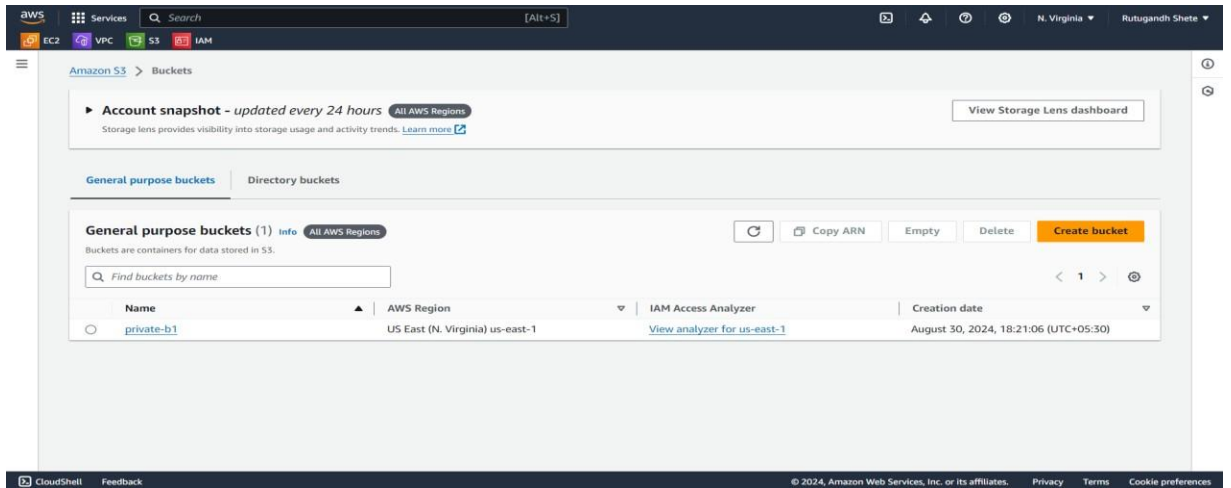
- In Bucket versioning → **Disabled:** The default state where versioning is not enabled, and **objects do not have version IDs.**  
**Enabled:** Once versioning is enabled, each object that you upload to the **bucket** receives a unique version ID.

### Bucket Versioning

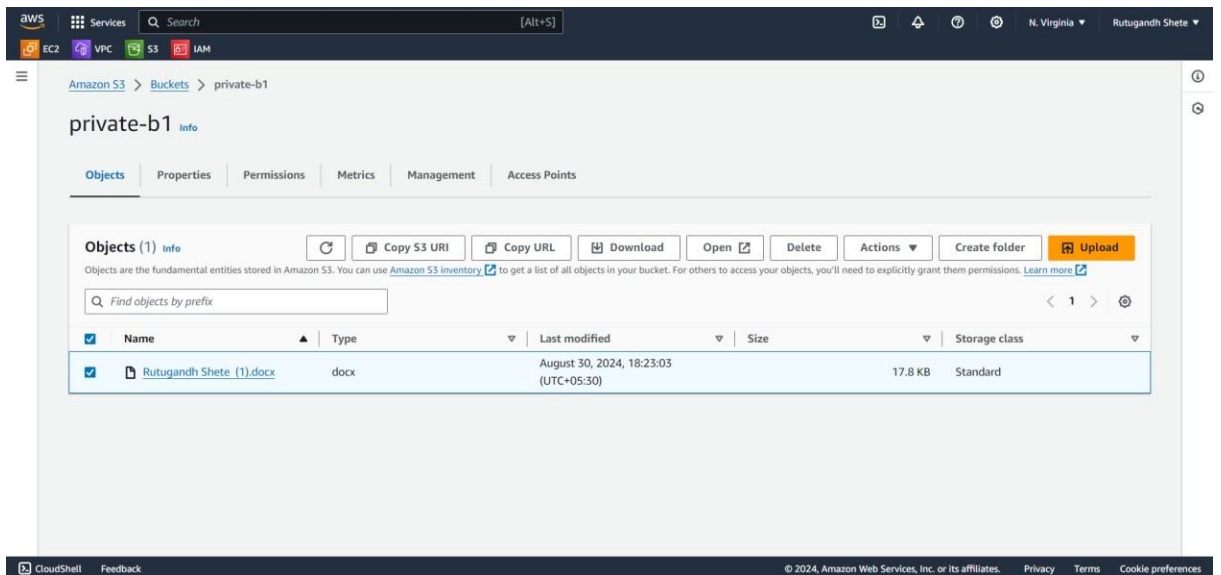
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.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

Bucket Versioning  
☒ Disable  
☐ Enable

- After creating bucket



- Add file into that bucket



- Copy URL and paste it into browser but we wont be able to access it publicly

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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>C398E78B9E68E5</RequestId>
  <HostId>x0YFjrophbrZEXBkr8c+K03rZhbcsyGClSBH0B1K+mbymTlFSJA9y+G5vMaDda9lVPrEH4yY7c</HostId>
</Error>
```

## Create simple bucket:

### Private-Bucket

#### Steps:

- Create bucket → Name → bucket type

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  
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](#)  
public-b2  
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

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 determines who can specify access to objects.

- Mention bucket ownership → **ACL's enabled** for making it public.

#### 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 determines who can specify access to objects.

☐ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.



We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

- **Uncheck this block**

#### Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ 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

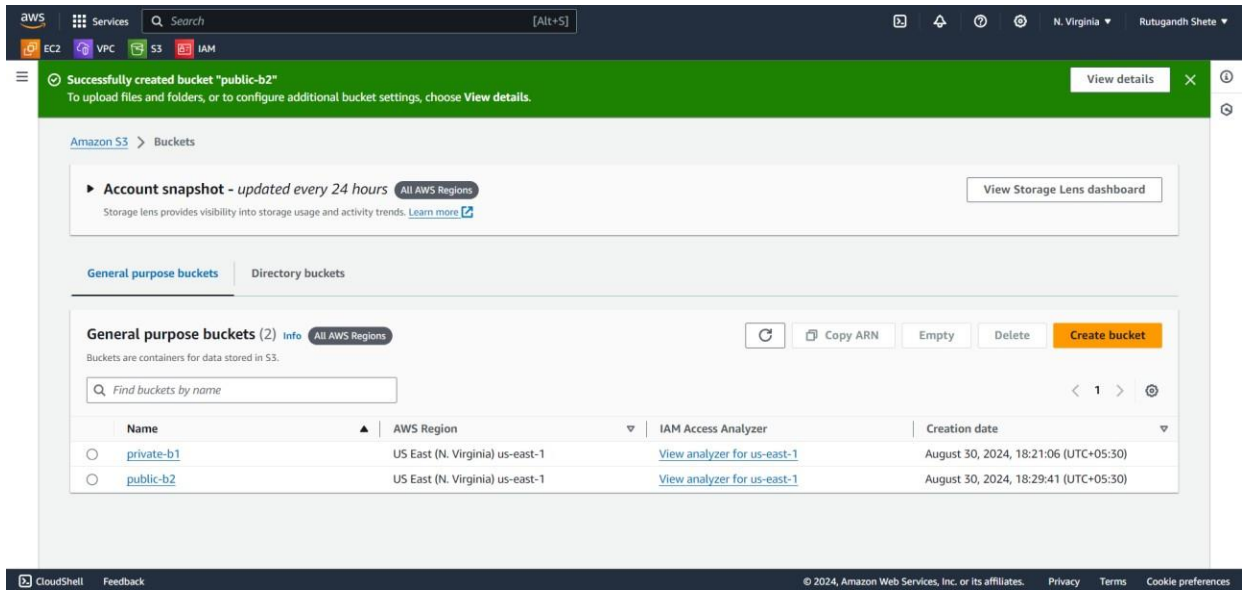
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through **any** public bucket or access point policies

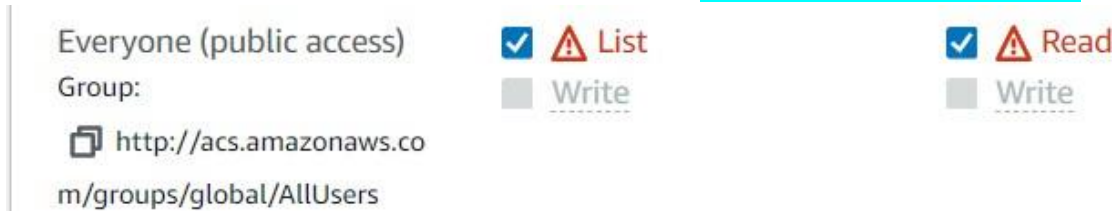
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



- After creating it



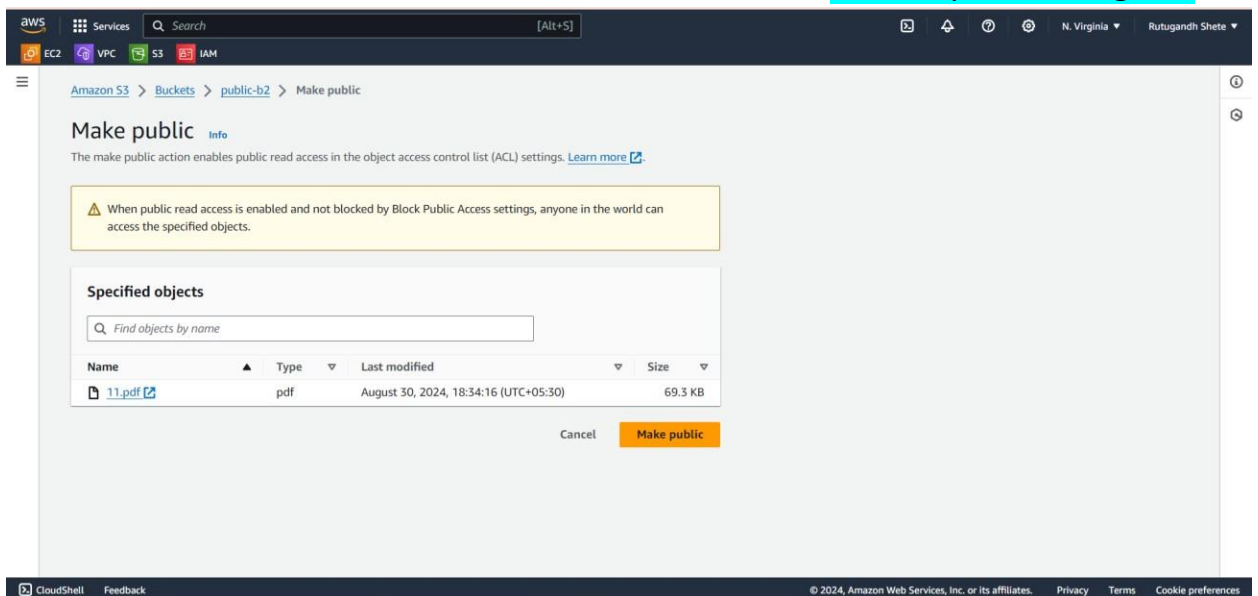
- Click on bucket → permissions → ACL's → edit → **check box to list and read.**



- We can give access to other AWS account



- Add one file into bucket → click on that file → actions → **make it public using ACL**

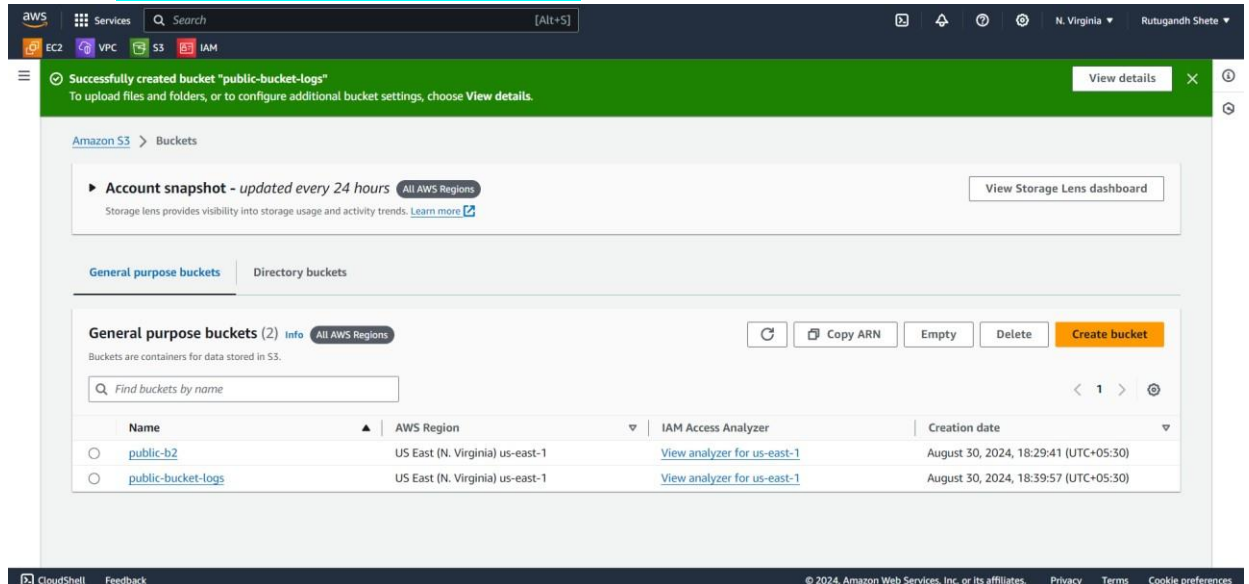




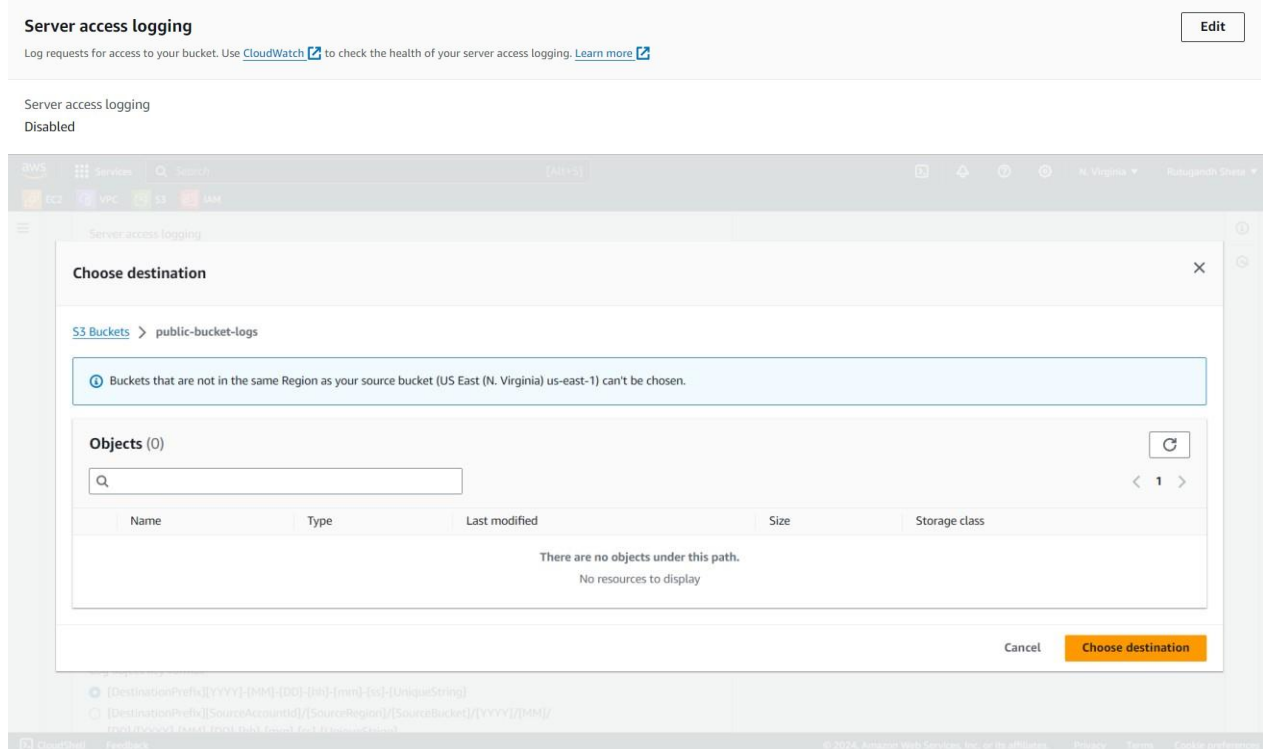
## Storing logs into another bucket.

### Steps:

- Create another bucket for storing logs into that bucket.



- Select bucket that we are going to check logs → properties → server logging → edit → enable → select bucket into which we are going to put all the logs





## Versioning of bucket

### Steps:

- Create bucket while creating it mention bucket **versioning enable** → add file into it  
→ click on that object and **toggle show versions**.

The image shows two screenshots from the AWS Management Console. The top screenshot displays the 'Edit Bucket Versioning' page for the bucket 'public-b2'. It shows the 'Bucket Versioning' section with the 'Enable' radio button selected. Below it, the 'Multi-factor authentication (MFA) delete' section is shown as 'Disabled'. The bottom screenshot shows the 'Objects' tab for the same bucket. It displays a table of objects with columns for Name, Type, Version ID, Last modified, Size, and Storage class. Two versions of 'version.txt' are listed.

**Edit Bucket Versioning**

**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

**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

**public-b2**

**Objects (2)**

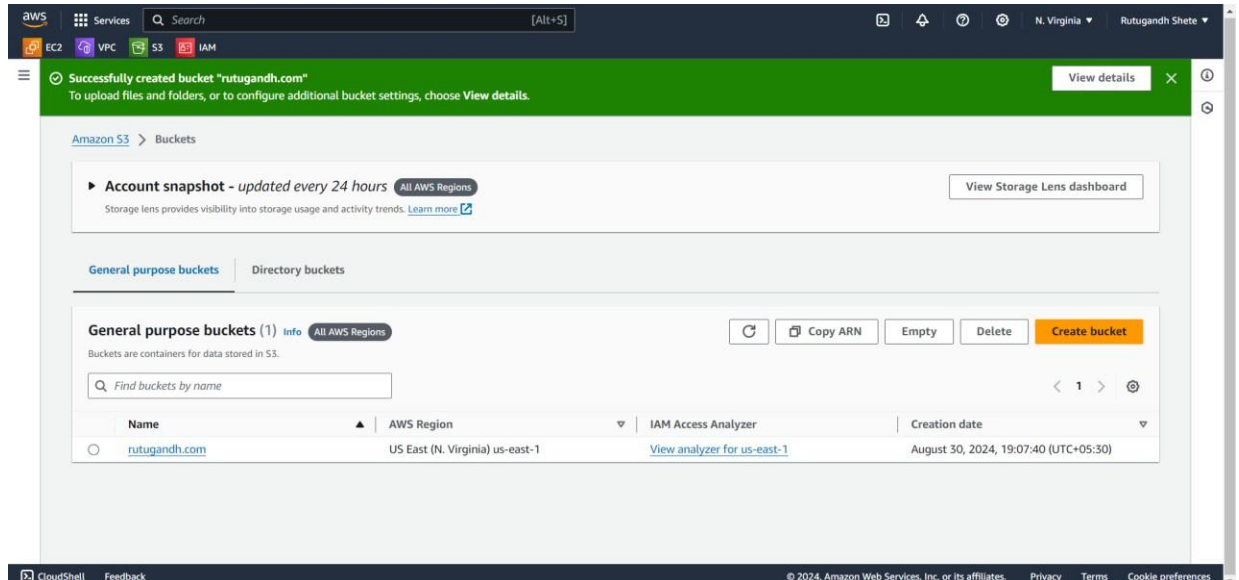
Find objects by prefix  Show versions ☒

	Name	Type	Version ID	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">version.txt</a>	txt	RfwFGAMot Sl5mVXSilUz 4m3V1_gVB4 R	August 30, 2024, 18:56:28 (UTC+05:30)	36.0 B	Standard
<input type="checkbox"/>	<a href="#">version.txt</a>	txt	7KFjyt53M8Q B3pljMek7ILv cuANZUpka	August 30, 2024, 18:54:05 (UTC+05:30)	16.0 B	Standard

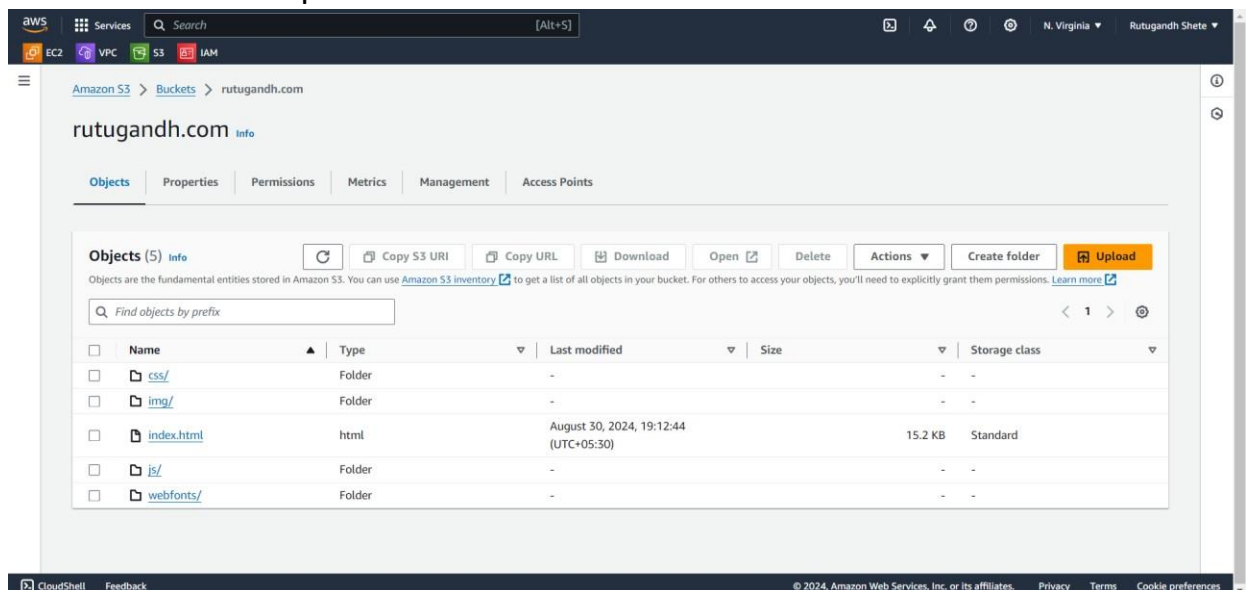
## Hosting one static website:

### Steps:

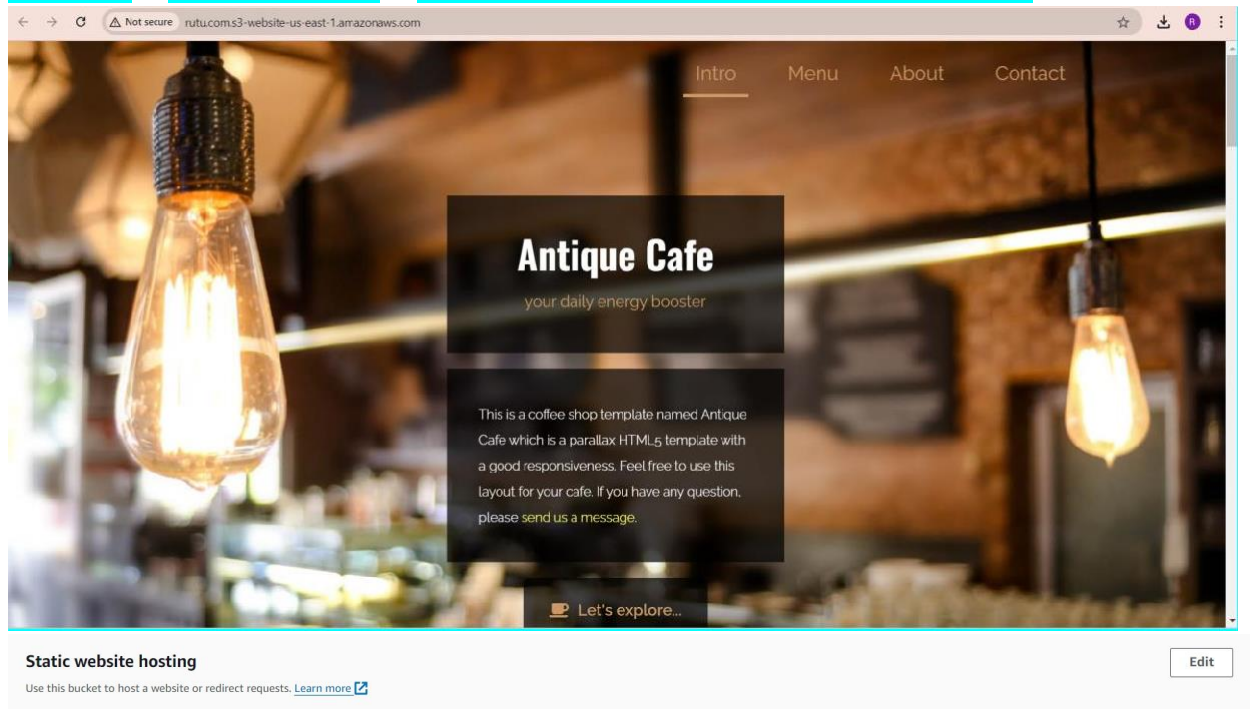
- Create bucket → ACL's is enabled → uncheck the public block → create bucket



- Download free template and all files into that bucket



- After adding files → go to **properties** → **edit** → **enable** → give html file name → **save changes** → **make it public** → **paste static website URL on the browser.**



## S3 storage classes.

### 1. General purpose

#### Amazon S3 Standard (S3 Standard)

S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.

Key features:

- General purpose storage for frequently accessed data
- Low latency and high throughput performance
- Designed to deliver 99.99% availability with an [availability SLA](#) of 99.9%

### 2. Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering)

The first cloud storage automatically decreases the user's storage cost. It provides very cost-effective access based on frequency, without affecting other performances. It also manages tough operations. Amazon S3 Intelligent – Tiering reduces the cost of granular objects automatically. No retrieval charges are there in Amazon S3 Intelligent – Tiering.

#### Characteristics of S3 Intelligent-Tiering

- Required less monitoring and automatically tier charge.
  - No minimum storage duration and no recovery charges are required to access the service.
  - Availability criteria are quite good like 99.9%.
  - Durability of S3 Intelligent- Tiering is 99.999999999%.\
- Amazon S3 Express One Zone

3. [Amazon S3 Express One Zone](#) is a high-performance, single-Availability Zone storage class purpose-built to deliver consistent single-digit millisecond data access for your most frequently accessed data and latency-sensitive applications. S3 Express One Zone can improve data access speeds by 10x and reduce request costs by 50% compared to S3 Standard. While you have always been able to choose a specific AWS Region to store your S3 data, with S3 Express One Zone you can select a specific AWS Availability Zone within an AWS Region to store your data. You can choose to co-locate your storage and compute resources in the same Availability Zone to further optimize performance, which helps lower compute costs and run workloads faster. With S3 Express One Zone, data is stored in a different bucket type—an Amazon S3 directory bucket—which supports hundreds of thousands of requests per second. Additionally, you can use S3 Express One Zone with services such as [Amazon SageMaker Model Training](#), [Amazon Athena](#), [Amazon EMR](#), and [AWS Glue](#) Data Catalog to accelerate your ML and analytics workloads. With S3 Express One Zone, storage automatically scales up or down based on your consumption and need, and you no longer need to manage multiple storage systems for low-latency workloads.

Key features:

- High performance storage for your most frequently accessed data
- Consistent single-digit millisecond request latency
- Improve access speeds by 10x and reduce request costs by 50% compared to S3 Standard
- Designed to deliver 99.95% availability with an [availability SLA](#) of 99.9%

#### 4. **S3 Standard-(IA) Infrequent Access: Cost-Effective Storage for Less Frequently Used Data**

To access the less frequently used data, users use S3 Standard-IA. It requires rapid access when needed. We can achieve high strength, high output, and low bandwidth by using S3 Standard-IA. It is best in storing the backup, and recovery of data for a long time. It acts as a data store for disaster recovery files.

##### **Identifying Suitable Data for S3 Standard-Infrequent Access**

To choose which type of data is suitable for the for S3 standard-infrequent access.

- Access Frequency
- Data Size
- Access Latency Requirements
- Data Durability Requirements

##### **Characteristics of S3 Standard-Infrequent Access**

- High performance and same action rate.
- Very Durable in all AZs.
- Availability is 99.9% in S3 Standard-IA.
- Durability is of 99.999999999%.

#### 5. **S3 Glacier Instant Retrieval: High-Performance Archiving with Rapid Retrieval**

It is an archive storage class that delivers the lowest-cost storage for data archiving and is organized to provide you with the highest performance and with more flexibility. S3 Glacier Instant Retrieval delivers the fastest access to archive storage. Same as in S3 standard, Data retrieval in milliseconds

##### **Characteristics of S3 Glacier Instant Retrieval**

- It just takes milliseconds to recover the data.
- The minimum object size should be 128KB.
- Availability is 99.9% in S3 glacier Instant Retrieval.
- Durability is of 99.999999999%.

#### 6. **S3 One Zone-Infrequent Access: Cost-Optimized Storage for Single Availability Zone**

Different from other S3 Storage Classes which store data in a minimum of three Availability Zones, S3 One Zone-IA stores data in a single Availability Zone and costs 20% less than S3 Standard-IA. It's a very good choice for storing secondary backup copies of on-premises data or easily re-creatable data. S3 One Zone-IA provides you the same high durability, high throughput, and low latency as in S3 Standard.

##### **Characteristics of S3 One Zone-Infrequent Access**

- Supports SSL (Secure Sockets Layer) for data in transferring and encryption of data.

- Availability Zone destruction can damage the data.
- Availability is 99.5% in S3 one Zone- Infrequent Access.
- Durability is of 99.999999999%.

## 7. S3 Glacier Flexible Retrieval: Balancing Cost and Retrieval Flexibility for Archiving

It provides low-cost storage compared to S3 Glacier Instant Retrieval. It is a suitable solution for backing up the data so that it can be recovered easily a few times in a year. It just takes minutes to access the data.

### Characteristics of S3 Glacier Flexible Retrieval

- Free recoveries in high quantity.
- AZs destruction can lead to difficulty in accessing data.
- when you have to retrieve large data sets, then S3 glacier flexible retrieval is best for backup and disaster recovery use cases.
- Availability is 99.99% in S3 glacier flexible retrieval.
- Durability is of 99.999999999%

## 8. Amazon S3 Glacier Deep Archive

The Glacier Deep Archive storage class is designed to provide long-lasting and secure long-term storage for large amounts of data at a price that is competitive with off-premises tape archival services that is very cheap. You no longer need to deal with expensive services. Accessibility is very much efficient, that it can restore data within 12 hours. This storage class is designed in such a way that users can easily get long-lasting and more secured storage for a huge amount of data at very less cost. Efficient accessibility and can restore data within very less time, therefore its time complexity is also efficient. S3 Glacier Deep Archive also have the feature of objects replication.

### Characteristics of S3 Glacier Deep Archive

- More secured storage.
- Recovery time is less requiring less time.
- Availability is 99.99% in S3 glacier deep archive.
- Durability is of 99.999999999%.



## Replication of bucket object:

### Steps:

- Create 2 s3 buckets one for source and another for destination, while creating s3 buckets ensure that we have enabled versioning. Name → enable versioning

### Bucket Versioning

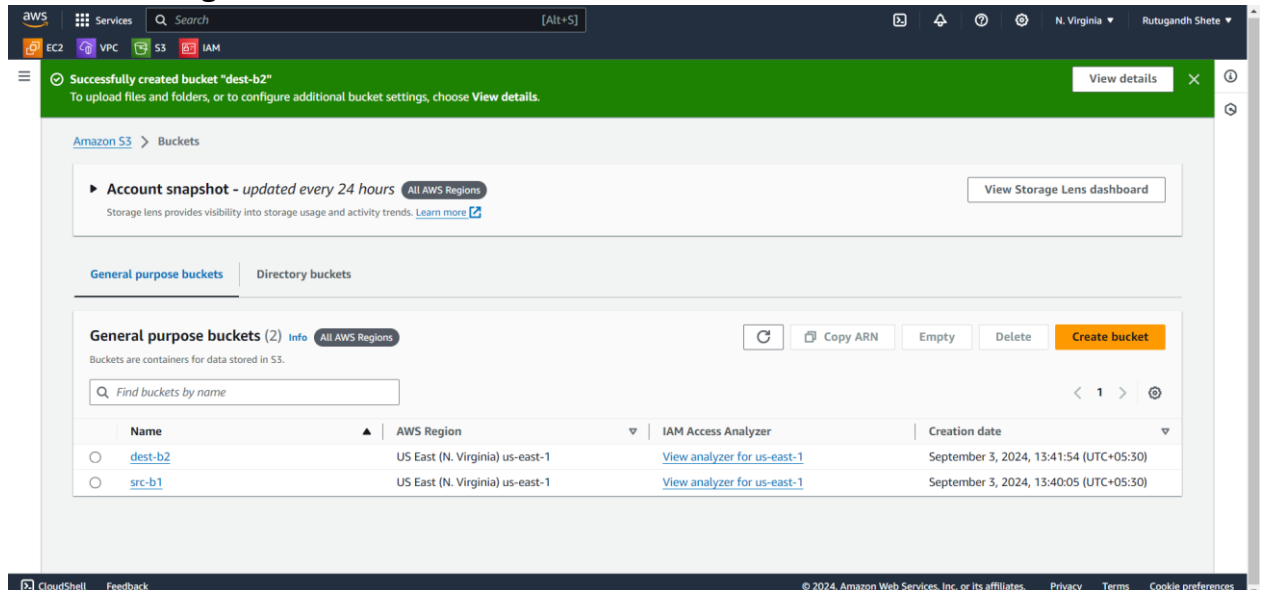
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.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

#### Bucket Versioning

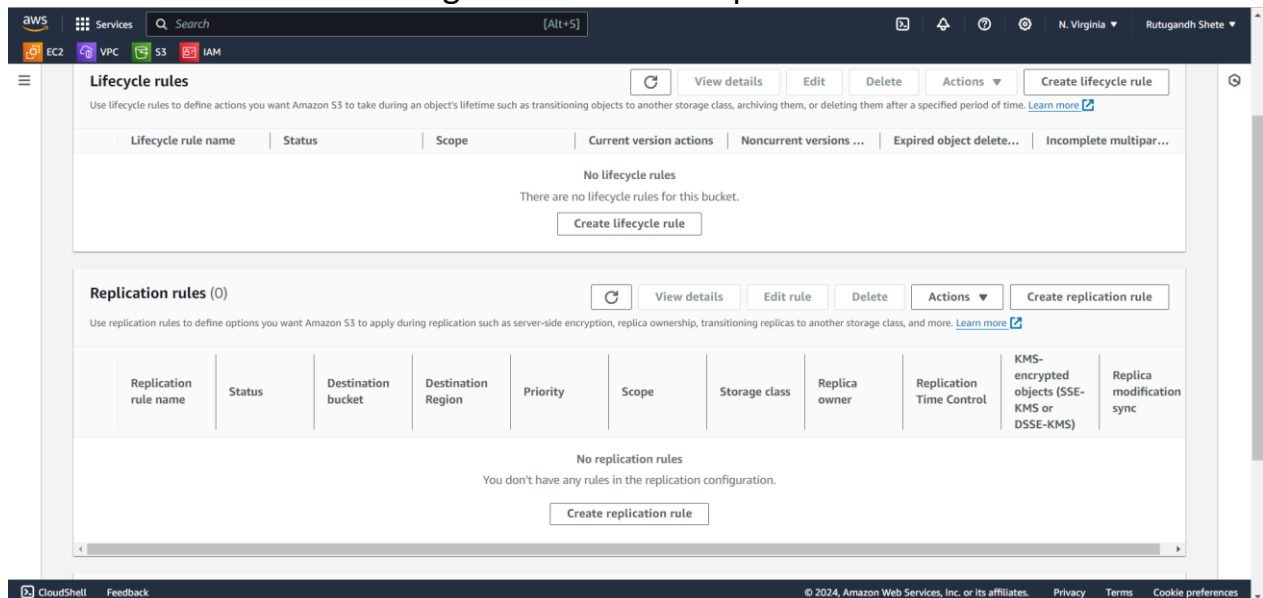
- ☒ Disable  
☐ Enable

- Create another bucket for destination. Name → enable versioning

- After creating 2 buckets.



- select source bucket → Management → create replication



- adding the source to bucket.

## Source bucket

Source bucket name

src-b1

Source Region

US East (N. Virginia) us-east-1

Choose a rule scope

- ☐ Limit the scope of this rule using one or more filters
- ☒ Apply to all objects in the bucket

- choose the destination

### Destination

Destination

You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#) or see [Amazon S3 pricing](#)

☒ Choose a bucket in this account

☐ Specify a bucket in another account

Bucket name

Choose the bucket that will receive replicated objects.

Browse S3

Destination Region

US East (N. Virginia) us-east-1

- set the IAM rule for replication

### IAM role

☒ Create new role

☐ Choose from existing IAM roles

☐ Enter IAM role ARN

- choose destination storage class.

aws

Services

Search

[Alt+S]

EC2

VPC

S3

IAM

N. Virginia

Rutugandh Shete

### Destination storage class

Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

☒ Change the storage class for the replicated objects

Storage class

	Storage class	Designed for	Availability Zones	Min storage duration	Min object size
<input checked="" type="radio"/>	Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
<input type="radio"/>	Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
<input type="radio"/>	Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1.
<input type="radio"/>	One Zone-IA	Recreateable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1.
<input type="radio"/>	Glacier Instant	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days	1.

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

27°C Cloudy

Search

ENG IN

14:37

03-09-2024

- after clicking on saving select whether you want to replicate existing object.

## Replicate existing objects?



You can enable a one-time Batch Operations job from this replication configuration to replicate objects that already exist in the bucket and to synchronize the source and destination buckets. [Learn more](#) or [see pricing](#)

### Existing objects

- ☒ No, do not replicate existing objects.
- ☐ Yes, replicate existing objects.

Cancel

Submit

The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information. Below the navigation bar, a green banner displays the message: 'Replication configuration successfully updated. If changes to the configuration aren't displayed, choose the refresh button. Changes apply only to new objects. To replicate existing objects with this configuration, choose Create replication job.' A 'Create replication job' button is visible on the right of the banner. Below the banner, the 'src-b1' source bucket is shown with its 'Source Region' as 'US East (N. Virginia) us-east-1'. A link to 's3rr\_role\_for\_src-b1' is provided. The main section is titled 'Replication rules (1)' and includes buttons for 'View details', 'Edit rule', 'Delete', and 'Actions'. A 'Create replication rule' button is also present. Below this, a table lists the replication rules. The table has columns for 'Replication rule name', 'Status', 'Destination bucket', 'Destination Region', 'Priority', 'Scope', 'Storage class', 'Replica owner', 'Replication Time Control', 'KMS-encrypted objects (SSE-KMS or DSSE-KMS)', and 'Replica modification sync'. The table contains one rule named 'replication-1' which is 'Enabled', with destination bucket 's3://dest-b2', region 'US East (N. Virginia) us-east-1', priority '0', scope 'Entire bucket', storage class 'Transition to Standard', replica owner 'Same as source', replication time control 'Disabled', KMS-encrypted objects 'Do not replicate', and replica modification sync 'Disabled'.

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted objects (SSE-KMS or DSSE-KMS)	Replica modification sync
<a href="#">replication-1</a>	Enabled	<a href="#">s3://dest-b2</a>	US East (N. Virginia) us-east-1	0	Entire bucket	Transition to Standard	Same as source	Disabled	Do not replicate	Disabled

## Create life-cycle rule:

Steps:

- Lifecycle name → select lifecycle rule scope.

The screenshot shows the AWS Management Console interface for creating a lifecycle rule. The breadcrumb navigation at the top reads: Amazon S3 > Buckets > src-b1 > Lifecycle configuration > Create lifecycle rule. The main heading is 'Create lifecycle rule' with an 'Info' link. Below this is the 'Lifecycle rule configuration' section. It contains a text input field for 'Lifecycle rule name' with the value 'lifecycle' and a note 'Up to 255 characters'. Under 'Choose a rule scope', the radio button 'Apply to all objects in the bucket' is selected. A yellow warning box states: 'Apply to all objects in the bucket. If you want the rule to apply to specific objects, you must use a filter to identify those objects. Choose "Limit the scope of this rule using one or more filters". Learn more'. A checkbox 'I acknowledge that this rule will apply to all objects in the bucket.' is checked. Below this is the 'Lifecycle rule actions' section, which includes a note: 'Choose the actions you want this rule to perform. Per-request fees apply. Learn more or see Amazon S3 pricing'. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for 2024.

- Select lifecycle rule action → and also select the transitions period between them. **1.**

### 1. Transition Actions

**Purpose:** Move objects to a different storage class to save money.

**How It Works:** You set a rule that moves your data to a cheaper storage class after a certain number of days.

**Example:**

You have a file that is frequently accessed for the first 30 days.

After 30 days, you don't need to access it as often, so you set a rule to move it to the **Standard-IA** (Infrequent Access) storage class.

Later, after 90 days, you might move it to **Glacier** for long-term, low-cost storage.

### 2. Expiration Actions

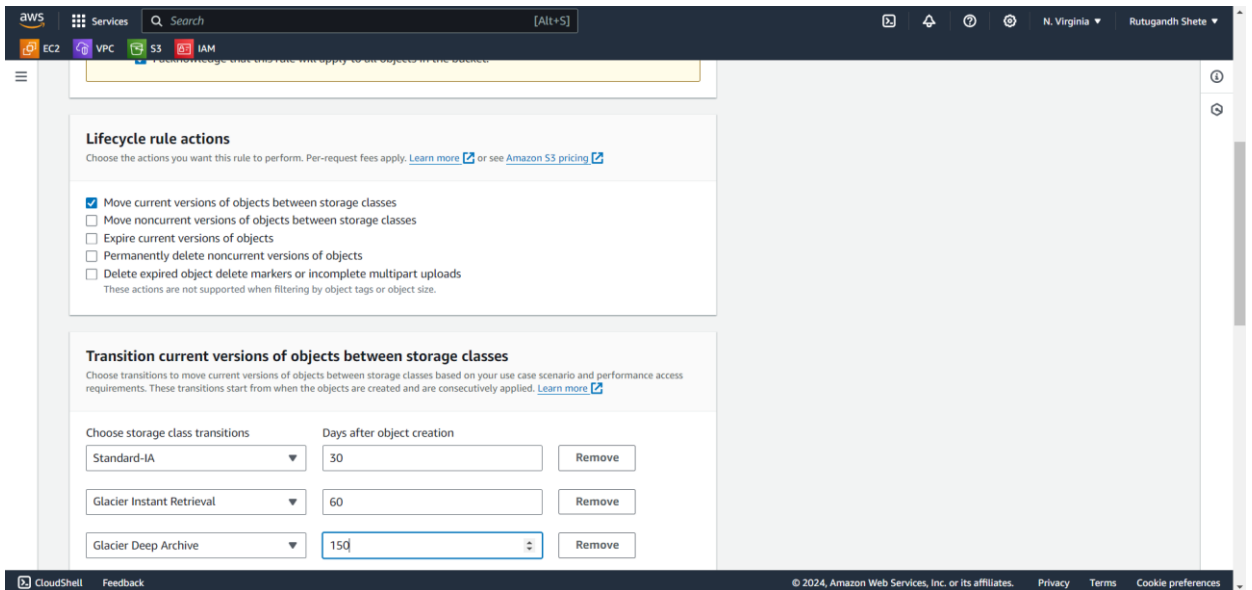
**Purpose:** Automatically delete objects or their versions after a certain time to free up space.

**How It Works:** You set a rule that deletes objects or old versions of them after a specified period.

**Example:**

You have log files that are only useful for 60 days.

You set a rule to delete these files after 60 days.



- To review all the transition

