

Amazon S3 and Glacier Storage

Amazon S3

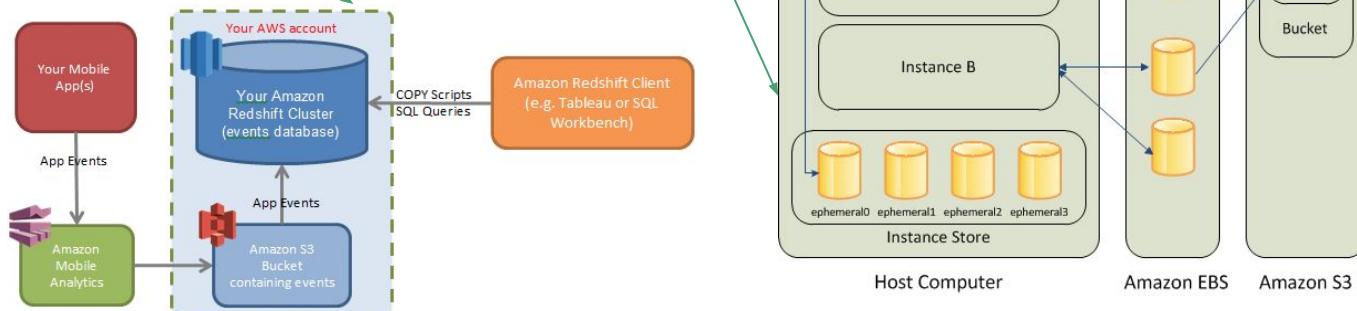
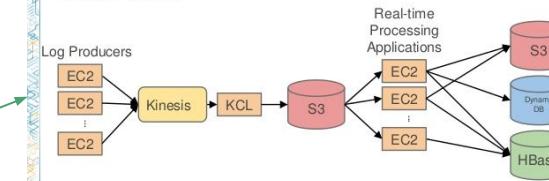
Amazon S3 is easy-to-use **object storage** with a simple **web service interface** that you can use to **store and retrieve any amount of data from anywhere** on the web. Amazon S3 also allows you to **pay only for the storage you actually use**, which eliminates the **capacity planning and capacity constraints** associated with traditional storage.

Amazon S3 can be used alone or in conjunction with other AWS services, and it offers a very high level of integration with many other AWS cloud services.

S3 inside AWS

- Amazon Kinesis
- Amazon Elastic MapReduce (Amazon EMR)
- Amazon Elastic Block Store (Amazon EBS)
- Amazon RDS snapshots
- Amazon Redshift
- Amazon DynamoDB
- And many more

Kinesis Pipeline Architecture
Flush to S3



Common use cases

- Backup and archive for on-premises or cloud data
- Content, media, and software storage and distribution
- Big data analytics
- Static website hosting
- Cloud-native mobile and Internet application hosting
- Disaster recovery

S3 Offerings

S3 offers a range of storage classes designed for various generic use cases:

- **General purpose**
- **Infrequent access**
- **Archive**

To help manage data through its lifecycle, Amazon S3 offers **configurable lifecycle policies**. By using lifecycle policies, you can have your data **automatically migrate to the most appropriate storage class**, without modifying your application code. In order to control who has access to your data, Amazon S3 provides a **rich set of permissions, access controls, and encryption options**.

Amazon Glacier

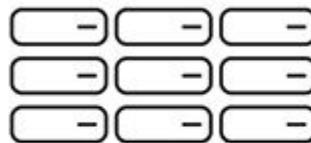
Another cloud storage service related to Amazon S3, but optimized for data archiving and long-term backup at extremely low cost.

Amazon Glacier is **suitable for “cold data”**, which is data that is **rarely accessed** and for which a **retrieval time of three to five hours is acceptable**. Amazon Glacier can be used both as a storage class of Amazon S3 (see Storage Classes and Object Lifecycle Management topics in the Amazon S3 Advanced Features section), and as an independent archival storage service (see the Amazon Glacier section).

Object Storage vs Traditional Block and File Storage

In traditional IT environments, two kinds of storage dominate:

- Block storage
- File storage



Block storage

Data stored in fixed-size ‘blocks’ in a rigid arrangement—ideal for enterprise databases



File storage

Data stored as ‘files’ in hierarchically nested ‘folders’—ideal for active documents



Object storage

Data stored as ‘objects’ in scalable ‘buckets’—ideal for unstructured big data, analytics and archiving

Block Storage Use Cases

- Block storage is ideal for databases, since a DB requires consistent I/O performance and low-latency connectivity.
- You can use block storage for RAID Volumes, where you combine multiple disks organized through striping or mirroring.
- Any application which requires service side processing, like Java, PHP, and .Net will require block storage.
- Running mission-critical applications like Oracle, SAP, Microsoft Exchange, and Microsoft SharePoint.

Amazon S3

Amazon S3 object storage is something quite different. **Amazon S3 is cloud object storage. Not tied or associated with a server**, Amazon S3 storage is independent of a server and is **accessed over the Internet**. Instead of managing data as blocks, data is **managed as objects using an Application Program Interface (API)** built on standard HTTP verbs.

GET an object or PUT an object

Advantages to Amazon S3

Amazon S3 is intentionally built with a minimal feature set that focuses on simplicity and robustness. Following are some of advantages of the Amazon S3 service:

- Create Buckets – Create and name a bucket that stores data. **Buckets are the fundamental container in Amazon S3** for data storage.
- Store data in Buckets – Store an **infinite amount of data in a bucket**. Upload as many objects as you like into an Amazon S3 bucket. Each **object can contain up to 5 TB of data**. Each **object is stored and retrieved using a unique developer-assigned key**.
- Download data – Download your data or enable others to do so. Download your data any time you like or allow others to do the same.

Advantages to Amazon S3

- Permissions – **Grant or deny access** to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms can help keep data secure from unauthorized access.
- Standard interfaces – Use **standards-based REST and SOAP interfaces** designed to work with any Internet-development toolkit.

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. It is recommended to use either the REST API or the AWS SDKs.

Making Requests

Amazon S3 is a REST service. You can send requests to Amazon S3 using the REST API or the AWS SDK wrapper libraries that wrap the underlying Amazon S3 REST API, simplifying your programming tasks.

Every interaction with Amazon S3 is either authenticated or anonymous. Authentication is a process of verifying the identity of the requester trying to access an Amazon Web Services (AWS) product. Authenticated requests must include a signature value that authenticates the request sender. The signature value is, in part, generated from the requester's AWS access keys (access key ID and secret access key).

If you are using the AWS SDK, the libraries compute the signature from the keys you provide. However, if you make direct REST API calls in your application, you must write the code to compute the signature and add it to the request.

- [Making Requests to Amazon S3 over IPv6](#)
- [Making Requests Using the AWS SDKs](#)
- [Making Requests Using the REST API](#)

Making Requests Using the AWS SDKs

You can send authenticated requests to Amazon S3 using either the AWS SDK or by making the REST API calls directly in your application. The AWS SDK API uses the credentials that you provide to compute the signature for authentication. If you use the REST API directly in your applications, you must write the necessary code to compute the signature for authenticating your request.

```
AmazonS3 s3client = new AmazonS3Client(new ProfileCredentialsProvider());
// Send sample request (list objects in a given bucket).
ObjectListing objectListing = s3client.listObjects(new
    ListObjectsRequest().withBucketName(bucketName));
```

Buckets

A bucket is a **container (web folder) for objects (files)** stored in Amazon S3. Every Amazon S3 object is contained in a bucket. Buckets form the **top-level namespace** for Amazon S3, and bucket names are global. This means that your **bucket names must be unique across all AWS accounts**, much like Domain Name System (DNS) domain names, not just within your own account. Bucket names can contain up to **63 lowercase letters, numbers, hyphens, and periods**. You can create and use multiple buckets; you can have up to **100 per account by default**.

For example, if the object named `photos/puppy.jpg` is stored in the `johndoe` bucket, then it is addressable using the URL
`http://johndoe.s3.amazonaws.com/photos/puppy.jpg`

Bucket Naming Rules

- Bucket names must be at **least 3 and no more than 63 characters** long.
- Bucket names must be a series of one or more labels. Adjacent labels are separated by a single period (.). Bucket names can contain lowercase letters, numbers, and hyphens. Each label must start and end with a lowercase letter or a number.
- Bucket names must not be formatted as an IP address (e.g., 192.168.5.4).
- It is recommended that you do not use periods (".") in bucket names.

Objects

Objects are the **entities or files stored in Amazon S3 buckets**. Range in size from **0 bytes up to 5TB**, a **single bucket can store an unlimited number of objects**. Each **object consists of data (the file itself) and metadata (data about the file)**. An **object's data is treated as simply a stream of bytes—Amazon S3 doesn't care what type of data you are storing**, and the service doesn't act differently for text data versus binary data.

- **System metadata** - date last modified, object size, MD5 digest, and HTTP Content-Type
- **User metadata** - optional and it can only be specified at the time an object is created. You can use custom metadata to tag your data with attributes that are meaningful to you.

Keys

Every object stored in an S3 bucket is identified by a unique identifier called a key. You can think of the key as a filename. **A key can be up to 1024 bytes of Unicode UTF-8 characters, including embedded slashes, backslashes, dots, and dashes.** **Keys must be unique within a single bucket**, but different buckets can contain objects with the same key. The combination of **bucket, key, and optional version ID uniquely identifies an Amazon S3 object.**

Object URL

Every Amazon S3 object can be addressed by a unique URL formed using the web services endpoint, the bucket name, and the object key.

ex: <http://mybucket.s3.amazonaws.com/jack.doc>

mybucket is the S3 bucket name, and jack.doc is the key or filename.

ex: <http://mybucket.s3.amazonaws.com/fee/fi/fo/fum/jack.doc>

mybucket is the S3 bucket name, fee/fi/fo/fum/jack.doc is the key.

A key may contain delimiter characters like slashes or backslashes to help you name and logically organize your Amazon S3 objects, but to Amazon S3 it is simply a long key name in a flat namespace. **There is no actual file and folder hierarchy.**

Regions

- **US East (N. Virginia) Region** Uses Amazon S3 servers in Northern Virginia
- **US East (Ohio) Region** Uses Amazon S3 servers in Columbus Ohio
- **US West (N. California) Region** Uses Amazon S3 servers in Northern California
- **US West (Oregon) Region** Uses Amazon S3 servers in Oregon
- **Canada (Central) Region** Uses Amazon S3 servers in Canada
- **Asia Pacific (Mumbai) Region** Uses Amazon S3 servers in Mumbai
- **Asia Pacific (Seoul) Region** Uses Amazon S3 servers in Seoul
- **Asia Pacific (Singapore) Region** Uses Amazon S3 servers in Singapore
- **Asia Pacific (Sydney) Region** Uses Amazon S3 servers in Sydney
- **Asia Pacific (Tokyo) Region** Uses Amazon S3 servers in Tokyo
- **EU (Frankfurt) Region** Uses Amazon S3 servers in Frankfurt
- **EU (Ireland) Region** Uses Amazon S3 servers in Ireland
- **EU (London) Region** Uses Amazon S3 servers in London
- **South America (São Paulo) Region** Uses Amazon S3 servers in São Paulo

Note

Objects stored in a region never leave the region unless you explicitly transfer them to another region.

Amazon S3 Operations

- Create/delete a bucket
- Write an object
- Read an object
- Delete an object
- List keys in a bucket

REST Interface

The native interface for Amazon S3 is a REST (Representational State Transfer) API. With the REST interface, you use standard HTTP or HTTPS requests to create and delete buckets, list keys, and read and write objects. REST maps standard HTTP “verbs” (HTTP methods) to the familiar CRUD (Create, Read, Update, Delete) operations. Create is HTTP PUT (and sometimes POST); read is HTTP GET; delete is HTTP DELETE; and update is HTTP POST (or sometimes PUT).

S3 Durability and Availability

Amazon S3 standard storage is designed for **99.99999999% durability and 99.99% availability** of objects over a given year.

For example, if you store 10,000 objects with Amazon S3, you can on average expect to incur **a loss of a single object once every 10,000,000 years**.

Amazon S3 achieves high durability by **automatically storing data redundantly on multiple devices in multiple facilities within a region**. It is designed to sustain the concurrent loss of data in two facilities without loss of user data. Amazon S3 provides a highly durable storage infrastructure designed for mission-critical and primary data storage.

Data Consistency

- Amazon S3 is an **eventually consistent system (read-after-write consistency)**
- Changes may **take some time to propagate to all locations**, immediately after an update may return stale data
- For PUTs to **new objects provides read-after write consistency**
- Amazon S3 offers eventual consistency for overwrite PUTS and Deletes in all regions.
- if you PUT to an existing key, a subsequent read might return the old data or the updated data, but **it will never write corrupted or partial data.**

Relook Data Consistency

- A process writes a new object to Amazon S3 and immediately lists keys within its bucket. **Until the change is fully propagated, the object might not appear in the list.**
- A process replaces an existing object and immediately attempts to read it. Until the change is fully propagated, **Amazon S3 might return the prior data.**
- A process deletes an existing object and immediately attempts to read it. Until the deletion is fully propagated, **Amazon S3 might return the deleted data.**
- A process deletes an existing object and immediately lists keys within its bucket. Until the deletion is fully propagated, **Amazon S3 might list the deleted object.**

Note

Amazon S3 does not currently support object locking. If two PUT requests are simultaneously made to the same key, the request with the latest timestamp wins.

Access Control on Objects

Amazon S3 is secure by default; when you create a bucket or object in Amazon S3, **only you have access**. The resource owner can optionally grant access permissions to others by writing an access policy. To allow you to give controlled access to others, Amazon S3 provides

- Coarse-grained access controls (Amazon S3 Access Control Lists [ACLs])
- Fine-grained access controls (Amazon S3 bucket policies, AWS Identity and Access Management [IAM] policies, and query-string authentication).

Managing Access

Covered in IAM

Amazon S3 ACLs

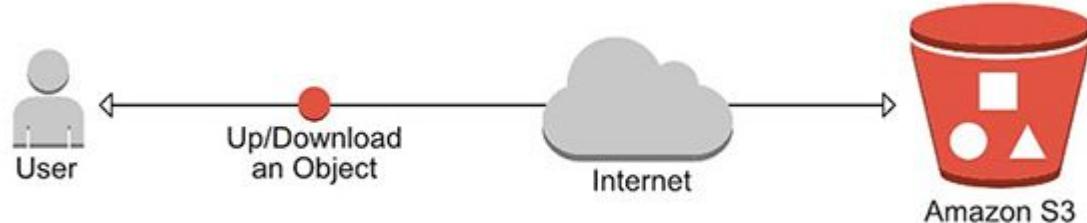
Amazon S3 ACLs allow you to **grant certain coarse-grained permissions: READ, WRITE, or FULL-CONTROL at the object or bucket level.** ACLs are a **legacy access control mechanism**, created before IAM existed. ACLs are best used today for a limited set of use cases, such as enabling bucket logging or making a bucket that hosts a static website be world-readable.

Amazon S3 bucket policies

Amazon S3 bucket policies are the recommended access control mechanism for Amazon S3 and provide much finer-grained control. Amazon S3 bucket policies are very similar to IAM policies but are subtly different in that:

S3 Key Points

- Highly-durable
- Highly-scalable
- Optimized for reads
- Minimalistic feature set
- Virtually any kind of data in any format
- Objects can range in size from 0 bytes up to 5TB,
- Unlimited number of files
- Data durability or replication across availability zones
- Automatically replicated on multiple devices in multiple facilities within a region
- Automatically partitions buckets to support very high request rates and simultaneous access by many clients

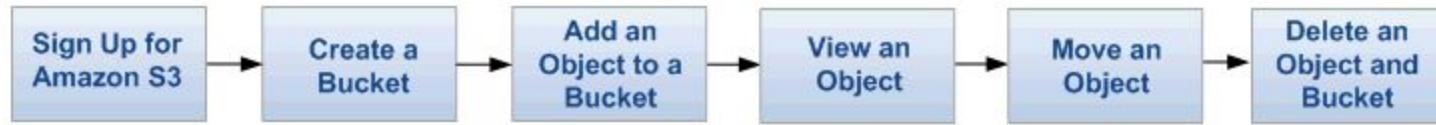


S3 Key Points

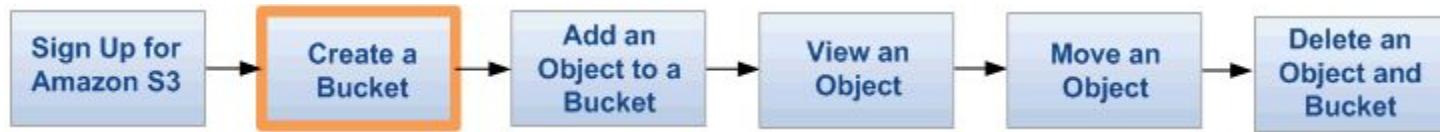
- S3 namespace is global but each Amazon S3 bucket is created in a specific region that you choose, this lets you control where your data is stored
- Bucket names can contain up to 63 lowercase letters, numbers, hyphens, and periods
- 100 buckets per account by default
- There is no file and folder hierarchy.

Practical: Create S3 Bucket

S3 Getting started



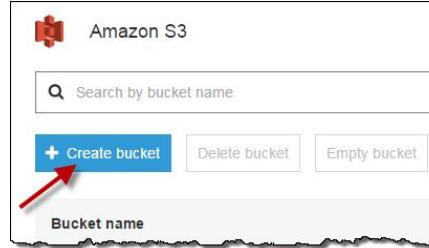
Create a Bucket



Create a Bucket

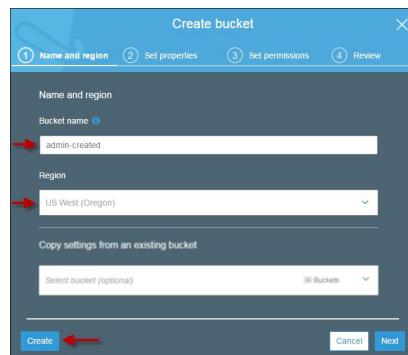
- Sign in to the AWS Management Console and open the Amazon S3 console

-



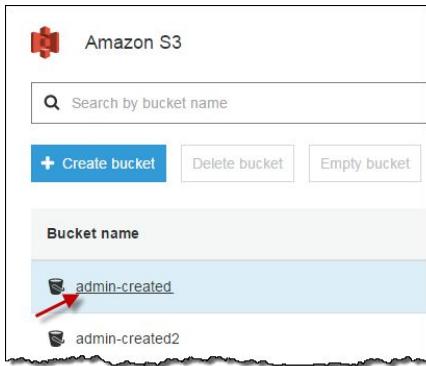
- In the **Bucket name** field, type a unique DNS-compliant name for your new bucket
- For **Region**, choose US West (Oregon) as the region where you want the bucket to reside.

-

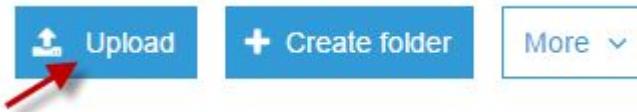


Add an Object to a Bucket

- In the **Bucket name** list, choose the name of the bucket that you want to upload your object to.



- Choose **Upload**.



Add an Object to a Bucket

- Or select “Get started”

This bucket is empty. Upload new objects to get started.

The screenshot shows a light blue background with three main sections. The first section on the left has a white bucket icon and the text "Upload an object". Below it is a description: "Buckets are globally unique containers for everything that you store in Amazon S3." At the bottom are "Learn more" and "Get started" buttons. The middle section has a silhouette of two people with a plus sign icon and the text "Set object properties". Below it is a description: "After you create a bucket, you can upload your objects (for example, your photo or video files)." At the bottom are "Learn more" and "Get started" buttons. The third section on the right has a database icon with a gear icon and the text "Set object permissions". Below it is a description: "By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others." At the bottom are "Learn more" and "Get started" buttons. A large central "Get started" button is positioned at the bottom center of the page.

Upload an object

Buckets are globally unique containers for everything that you store in Amazon S3.

[Learn more](#)

Set object properties

After you create a bucket, you can upload your objects (for example, your photo or video files).

[Learn more](#)

Set object permissions

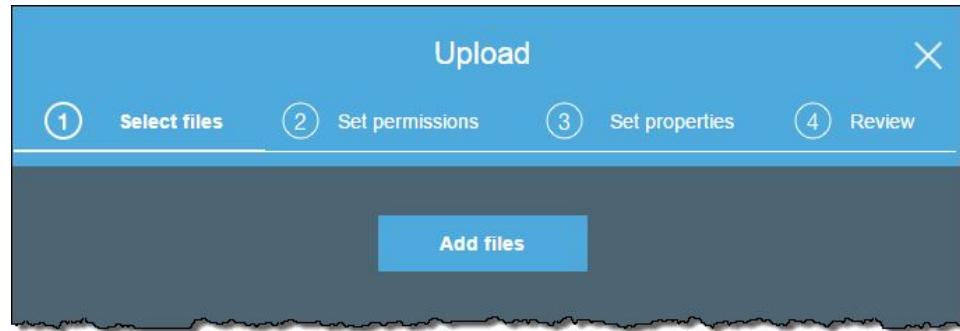
By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others.

[Learn more](#)

Get started

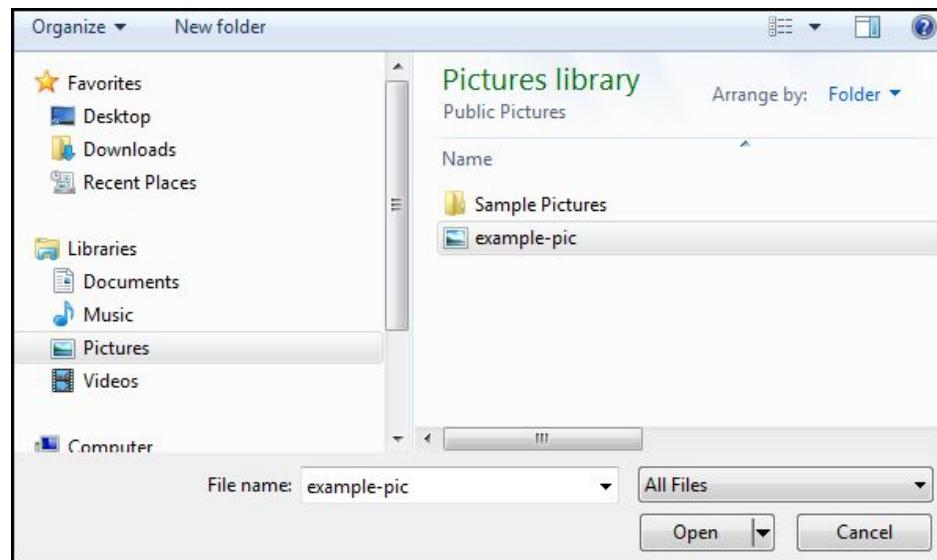
Add an Object to a Bucket

- In the **Upload** dialog box, choose **Add files** to choose the file to upload.



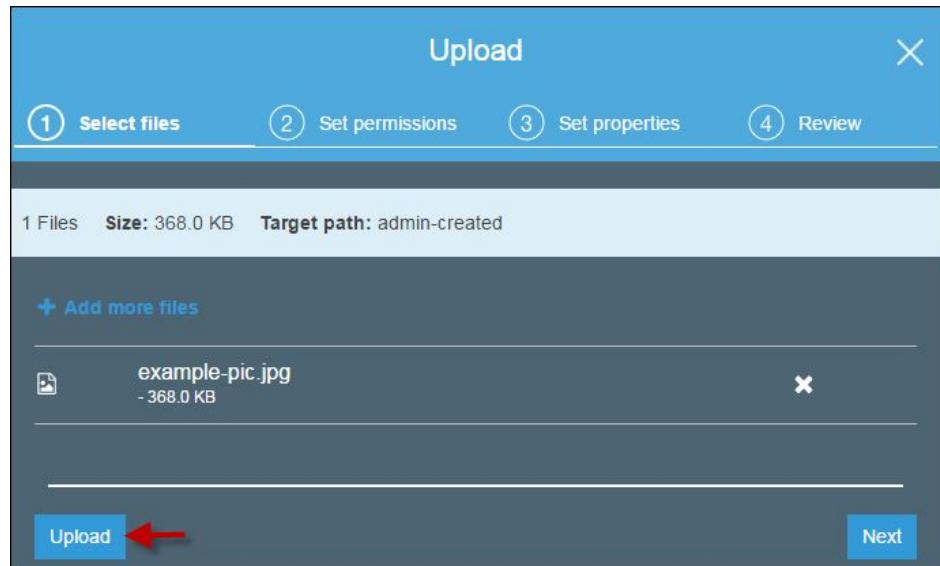
Add an Object to a Bucket

- Choose a file to upload, and then choose **Open**.



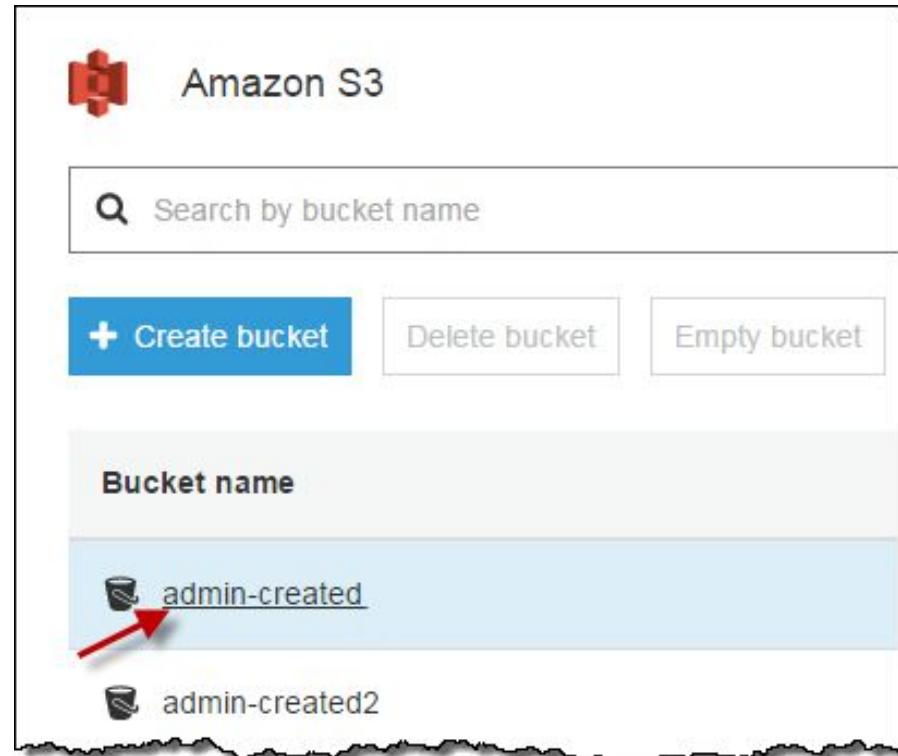
Add an Object to a Bucket

- Choose Upload.



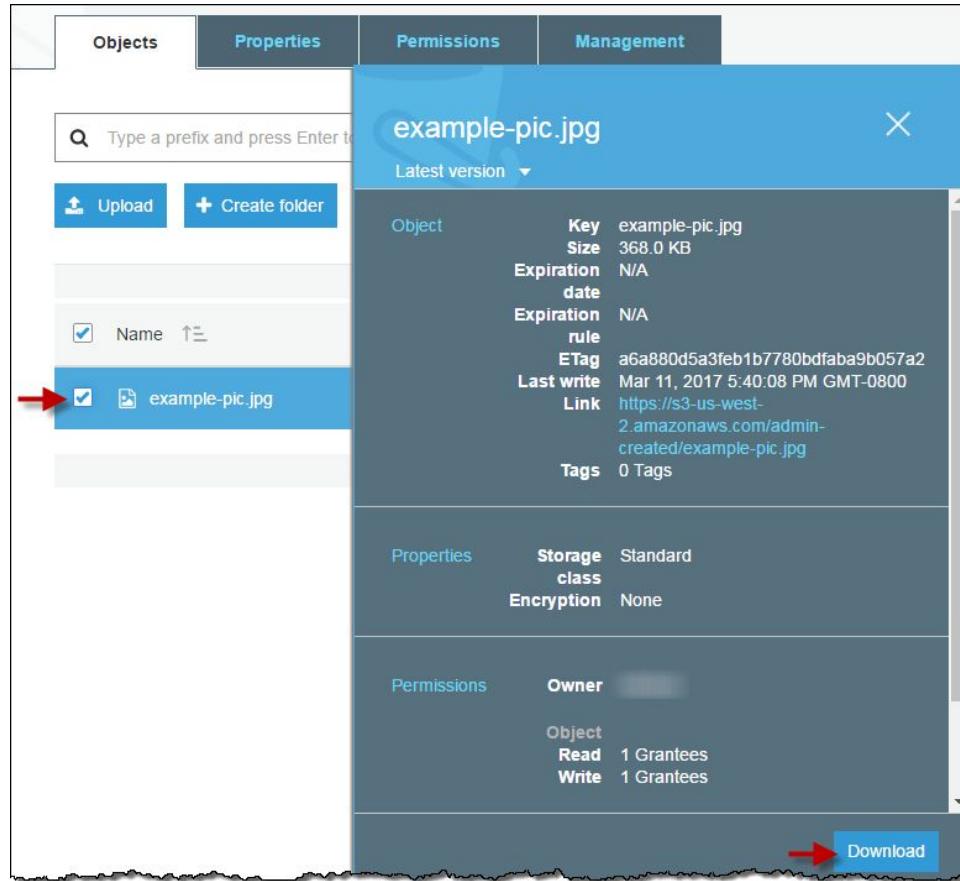
View an Object

- In the **Bucket name** list, choose the name of the bucket that you created.



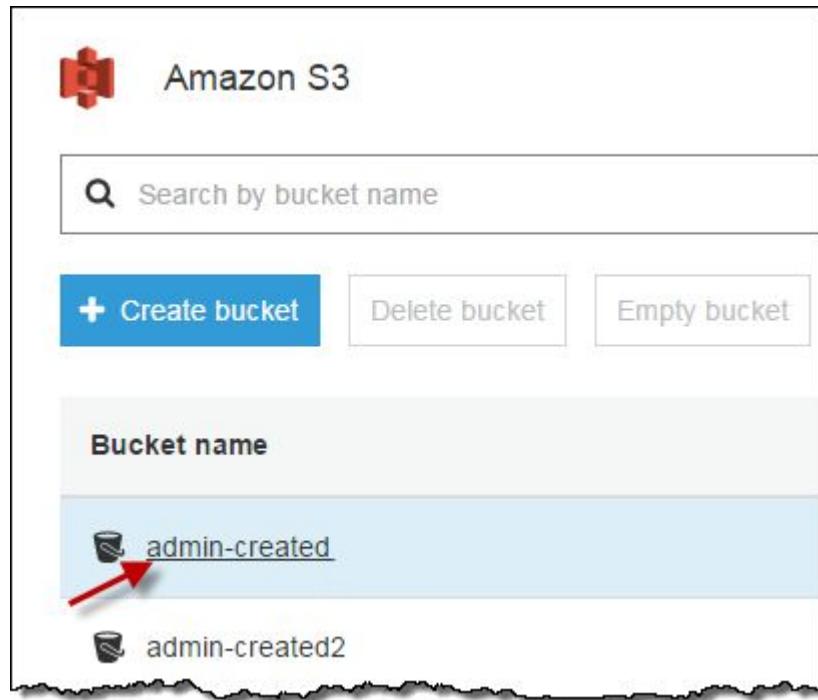
View an Object

- In the **Name** list, select the checkbox next to the object that you uploaded, and then choose **Download** on the object overview panel.



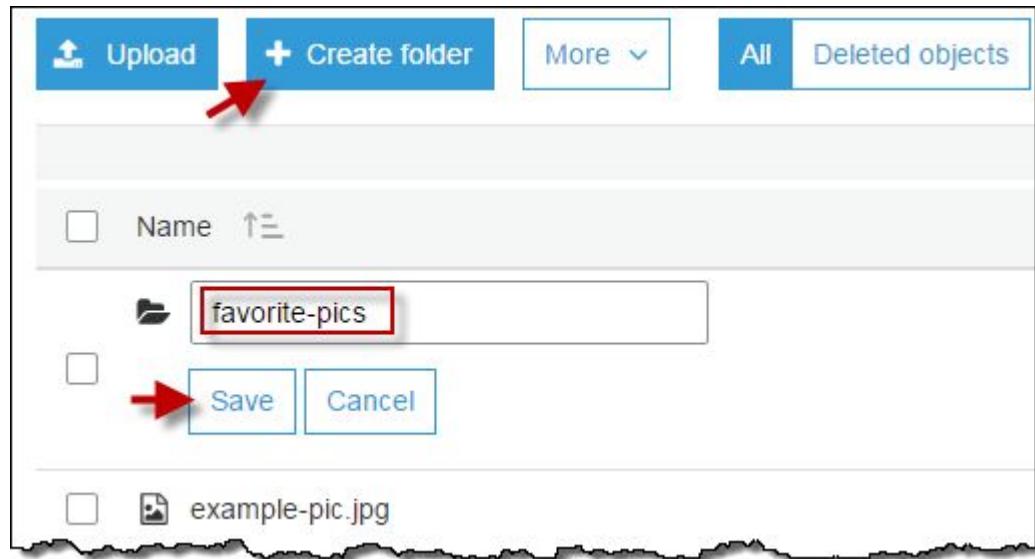
Move an Object

In the **Bucket name** list, choose the name of the bucket that you created.



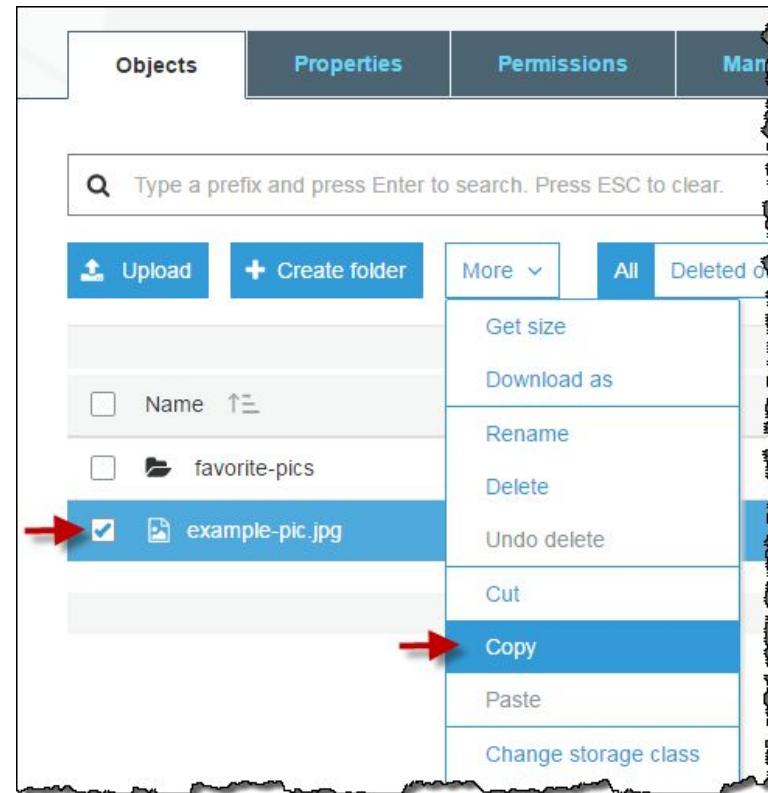
Move an Object

Choose **Create Folder**, type **favorite-pics** for the folder name, and then choose **Save**.



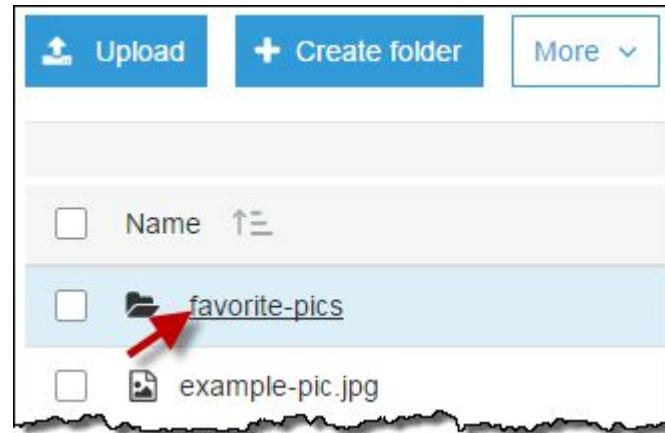
Move an Object

In the **Name** list, select the check box next to the object that you want to copy, choose **More**, and then choose **Copy**.



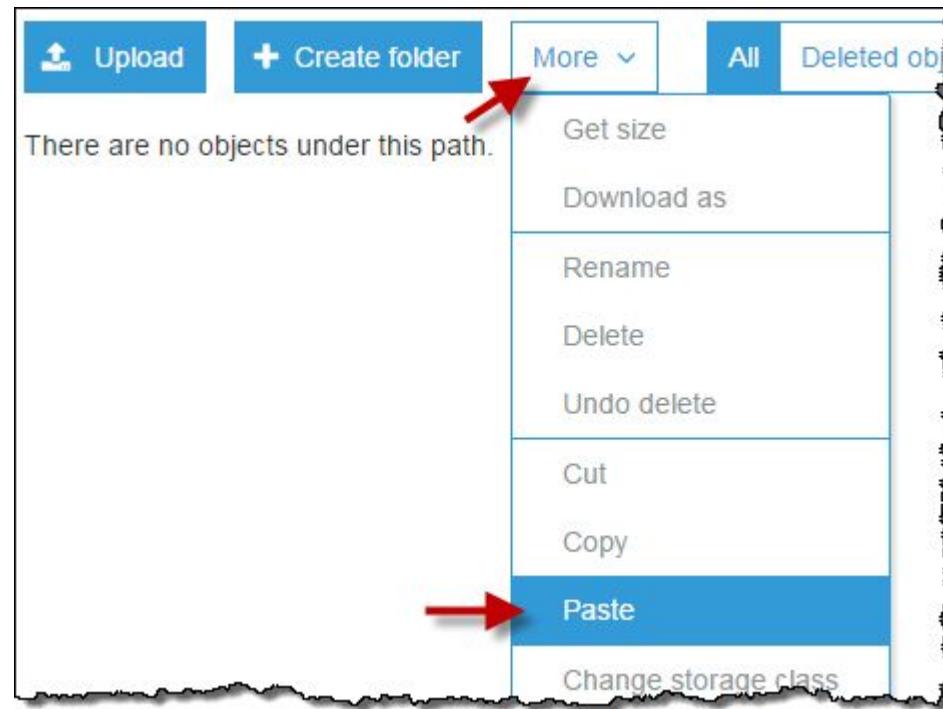
Move an Object

In the **Name** list, choose the name of the folder **favorite-pics**.



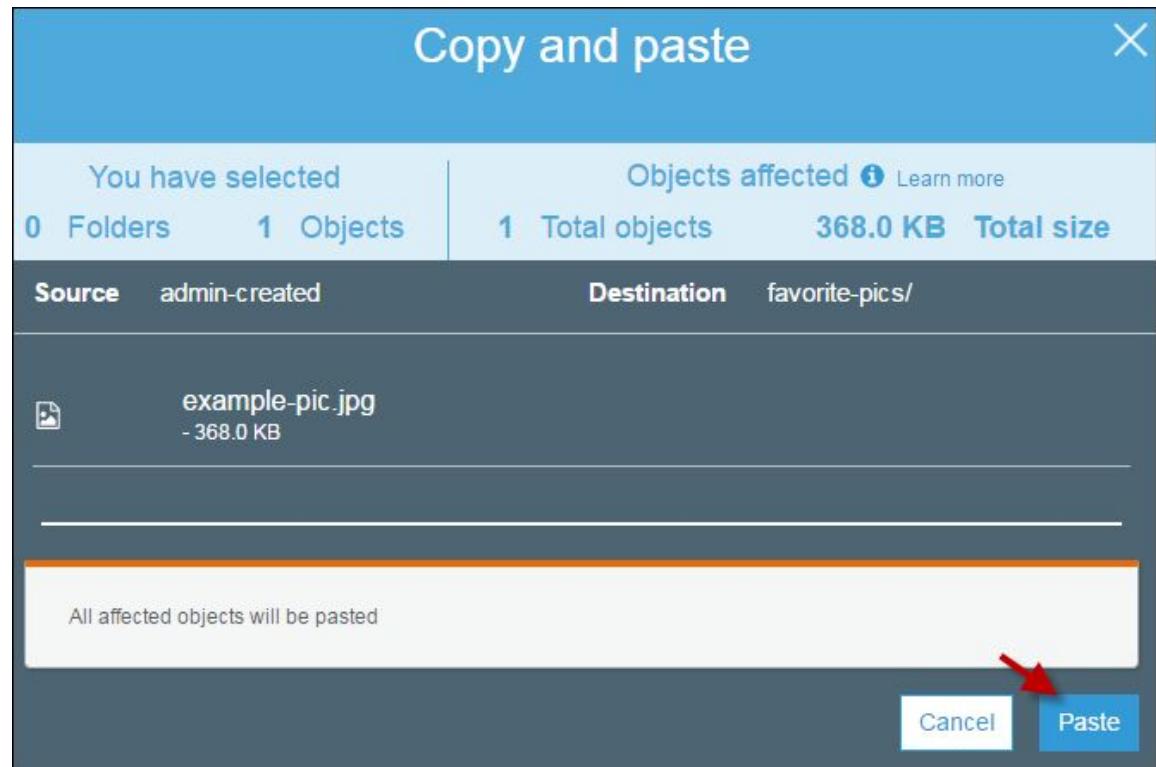
Move an Object

Choose **More**, and then choose **Paste**.



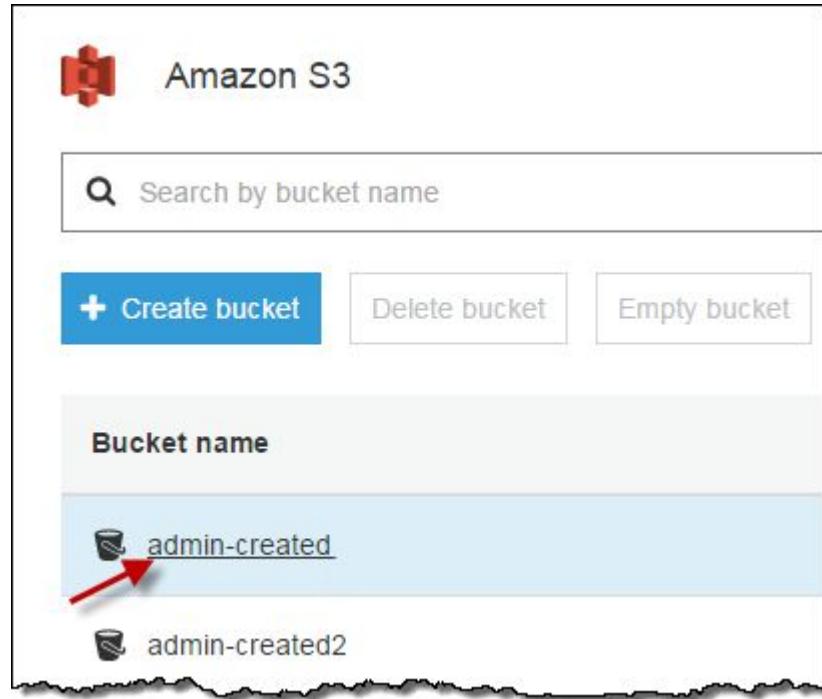
Move an Object

Choose Paste.



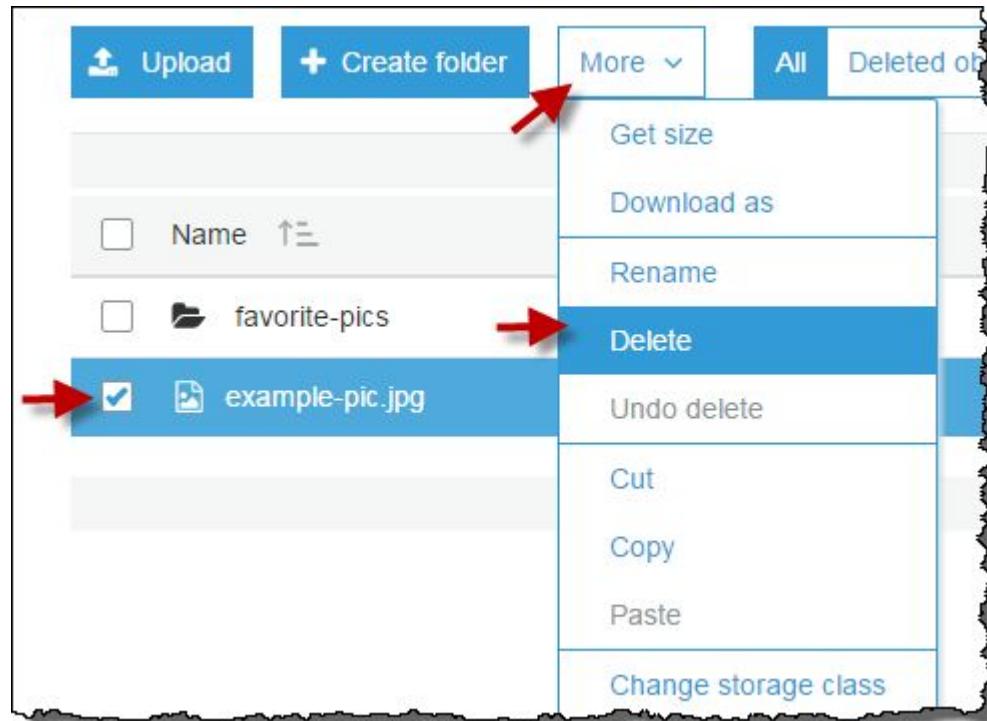
Delete an Object

In the **Bucket name** list, choose the name of the bucket that you want to delete an object from.



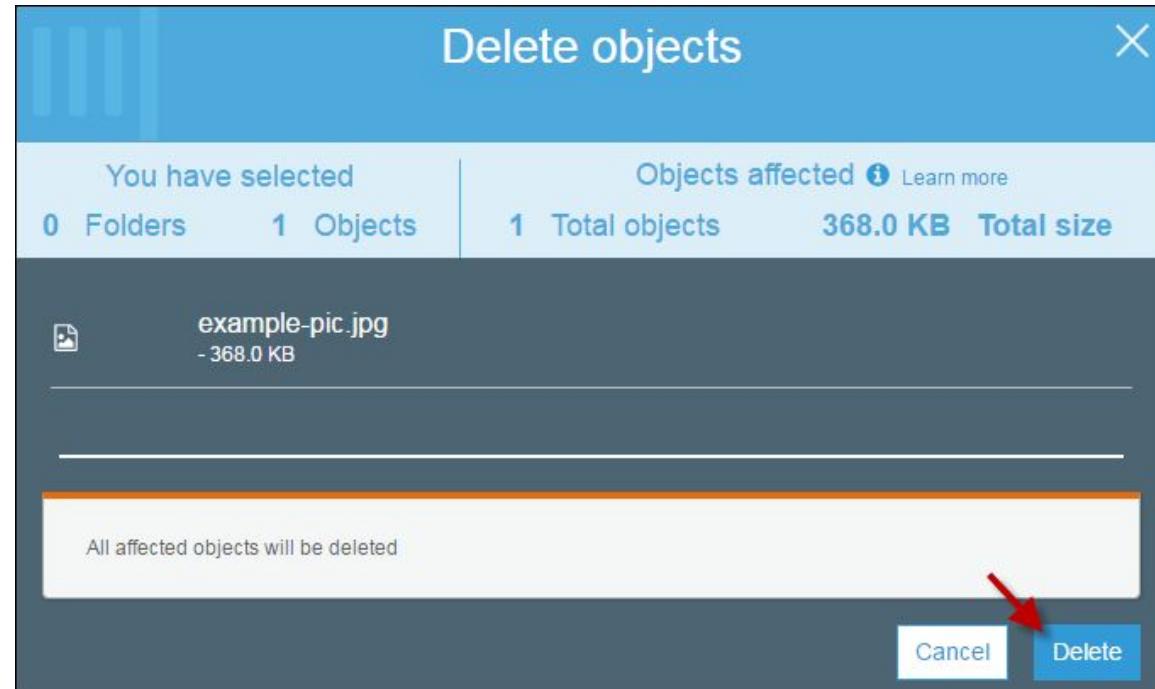
Delete an Object

In the **Name** list, select the checkbox next to the object that you want to delete, choose **More**, and then choose **Delete**.



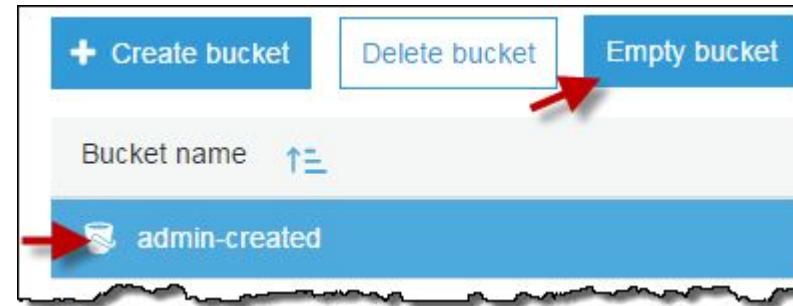
Delete an Object

In the **Delete objects** dialog box, verify that the name of the object you selected for deletion is listed, and then choose **Delete**.



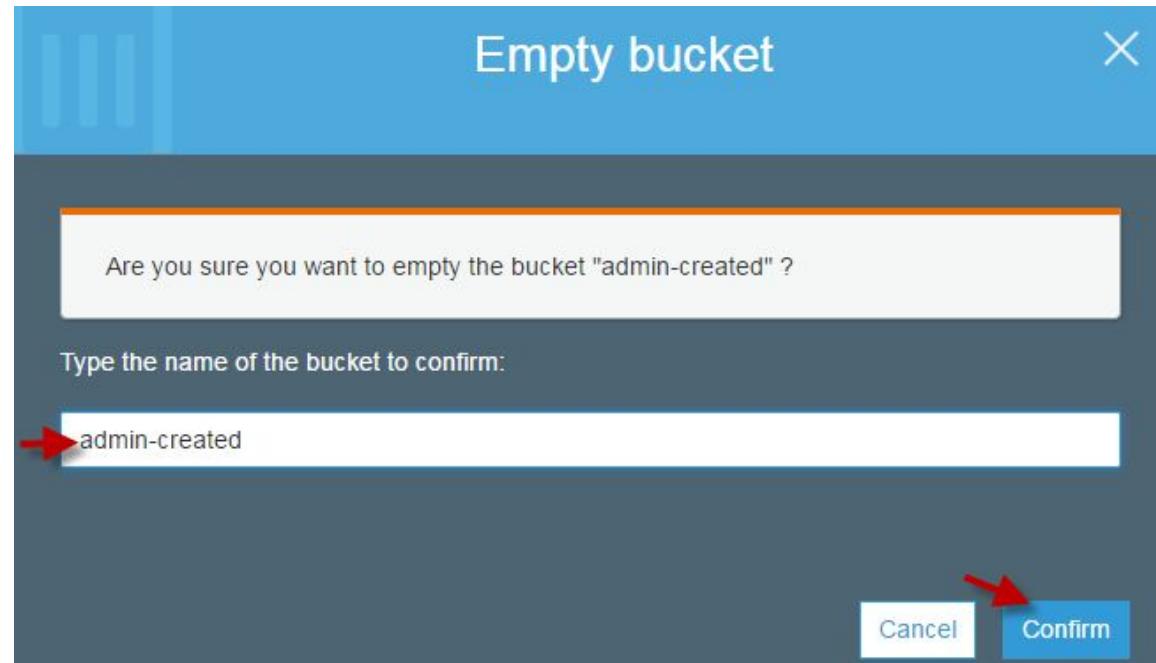
Empty a Bucket

In the **Bucket name** list, choose the bucket icon next to the name of the bucket that you want to empty and then choose **Empty bucket**.



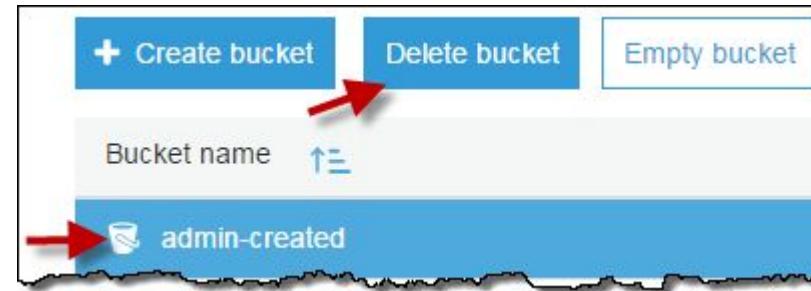
Empty a Bucket

In the **Empty bucket** dialog box, type the name of the bucket for confirmation and then choose **Confirm**.



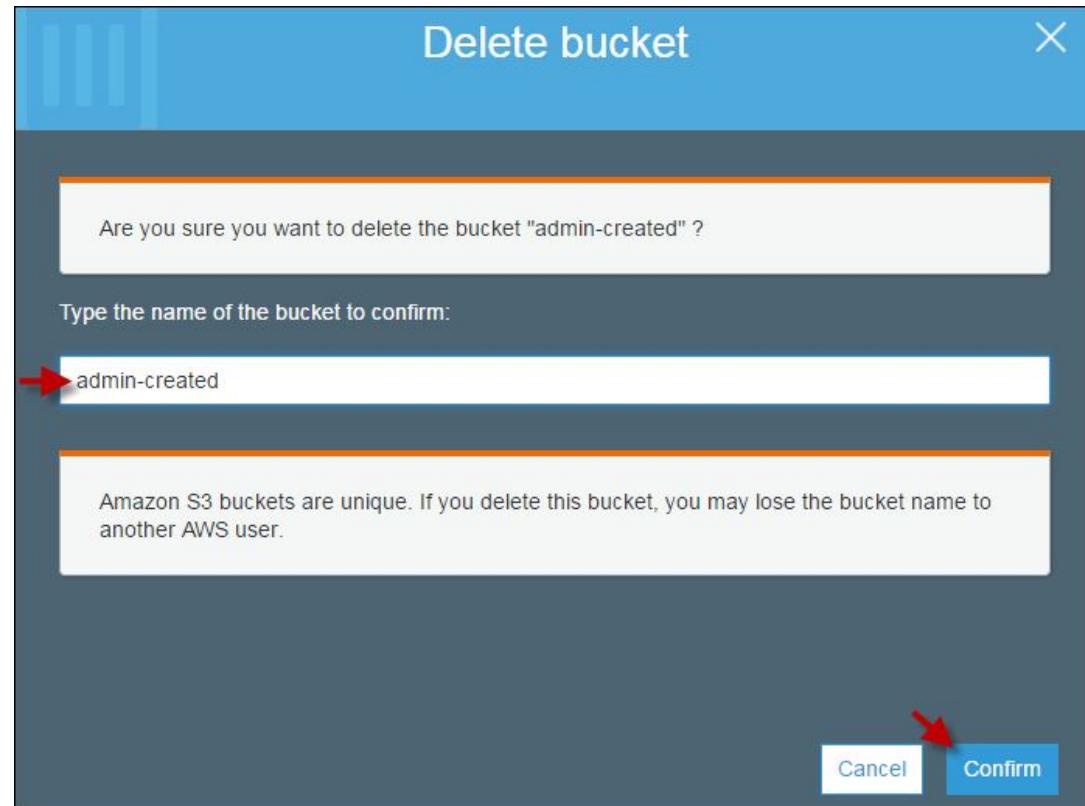
Delete a Bucket

In the **Bucket name** list, choose the bucket icon next to the name of the bucket that you want to delete and then choose **Delete bucket**.



Delete a Bucket

In the **Delete bucket** dialog box, type the name of the bucket for delete confirmation and then choose **Confirm**.



Practical: Create a File

Create a bucket

Create bucket

① Name and region ② Set properties ③ Set permissions ④ Review

Name and region

Bucket name i

Enter DNS-compliant bucket name

Region

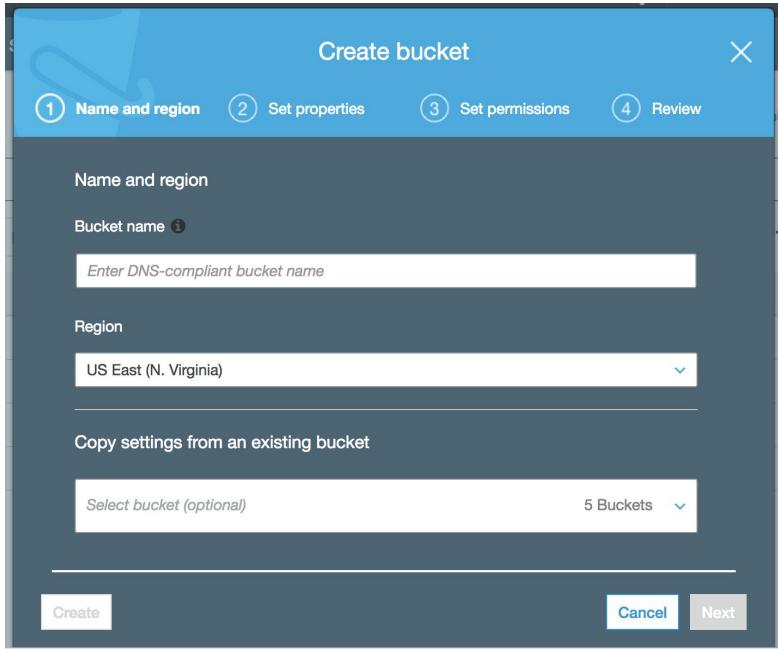
US East (N. Virginia)

Copy settings from an existing bucket

Select bucket (optional)

5 Buckets v

Create Cancel Next



Create bucket

① Name and region ② Set properties ③ Set permissions ④ Review

Name and region

Bucket name i

itskills-july-2017

Region

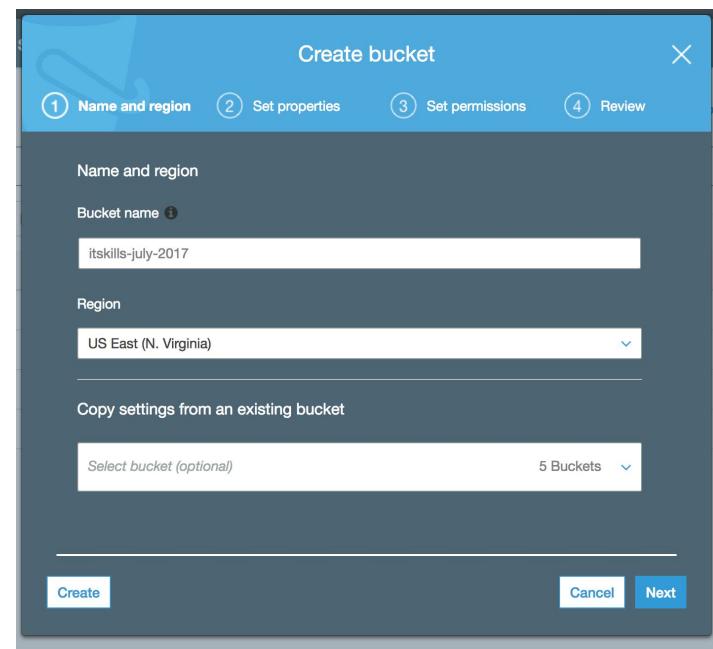
US East (N. Virginia)

Copy settings from an existing bucket

Select bucket (optional)

5 Buckets v

Create Cancel Next



Bucket Properties

Versioning

Enable versioning
 Suspend versioning
This suspends the creation of object versions for all operations but preserves any existing object versions.

[Cancel](#) [Save](#)

Tags

Key [i](#) Value [i](#)

[+ Add tag](#)

[Cancel](#) [Save](#)

Logging

Enable logging
 Disable logging

[Cancel](#) [Save](#)

Bucket Permissions

Manage users

User ID	Objects	Object permissions
rohit.kkumar(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write

X

Manage public permissions

Do not grant public read access to this bucket (Recommended) ▾

Do not grant public read access to this bucket (Recommended)

Grant public read access to this bucket

Manage Bucket Permissions

Manage system permissions

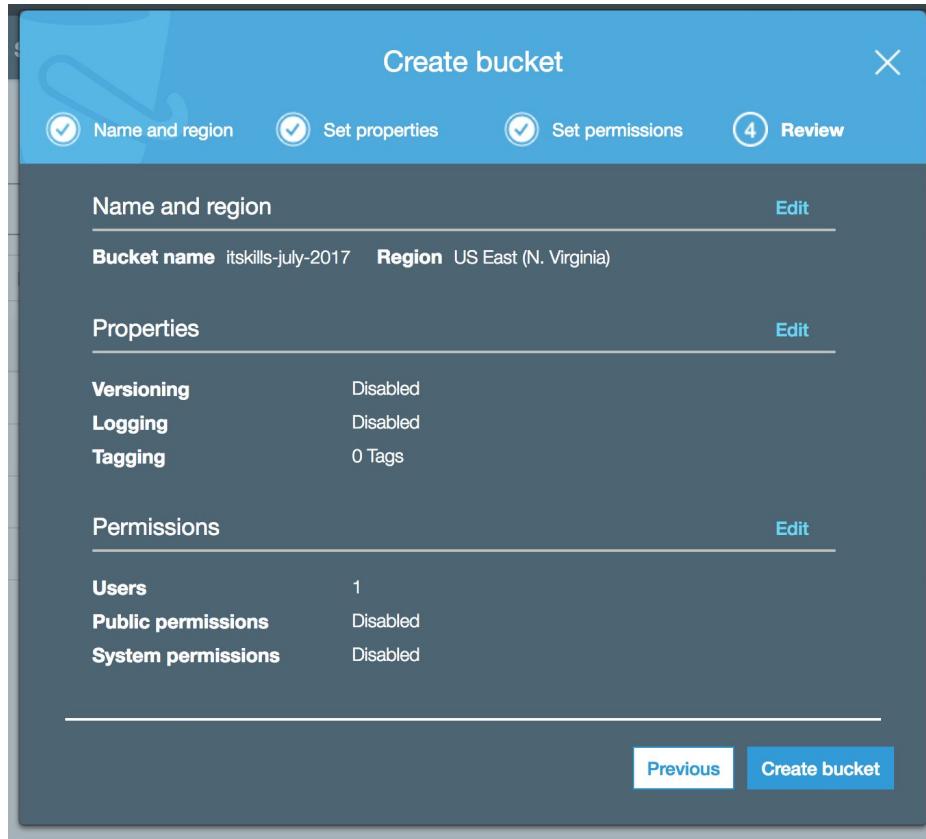
Do not grant Amazon S3 Log Delivery group write access to this bucket

Manage users

User ID	Objects	
rohit.kkumar(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write

Grant permissions to the user to read or write to an access control list (ACL) for the bucket.

Review Bucket & Create





Amazon S3 > itskills-july-2017

Overview

Properties

Permissions

Management

Versioning

Keep multiple versions of an object in the same bucket.

[Learn more](#)

 Disabled

Logging

Set up access log records that provide details about access requests.

[Learn more](#)

 Disabled

Static website hosting

Host a static website, which does not require server-side technologies.

[Learn more](#)

 Disabled

Advanced settings

Tags

Use tags to track your cost against projects or other criteria.

[Learn more](#)

 0 Tags

Cross-region replication

Automate copying objects across different AWS Regions.

[Learn more](#)

 Disabled

Transfer acceleration

Enable fast, easy and secure transfers of files to and from your bucket.

[Learn more](#)

 Suspended

Events

Receive notifications when specific events occur in your bucket.

[Learn more](#)

 0 Active notifications

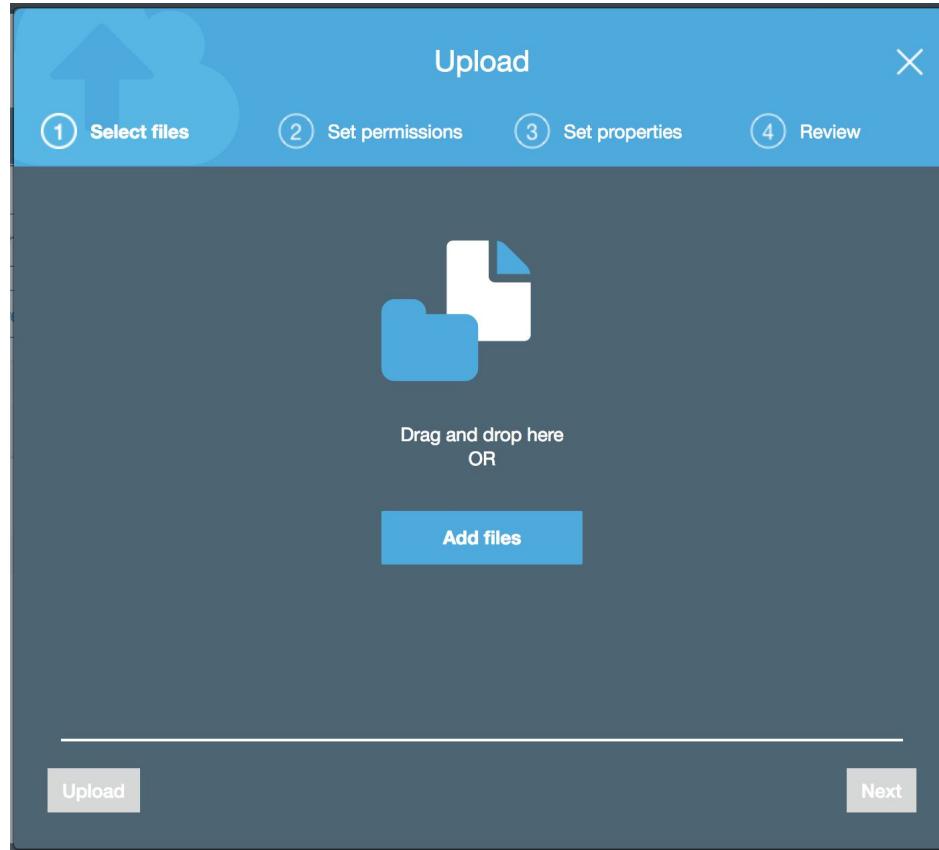
Requester pays

The requester (instead of the bucket owner) will pay for requests and data transfer.

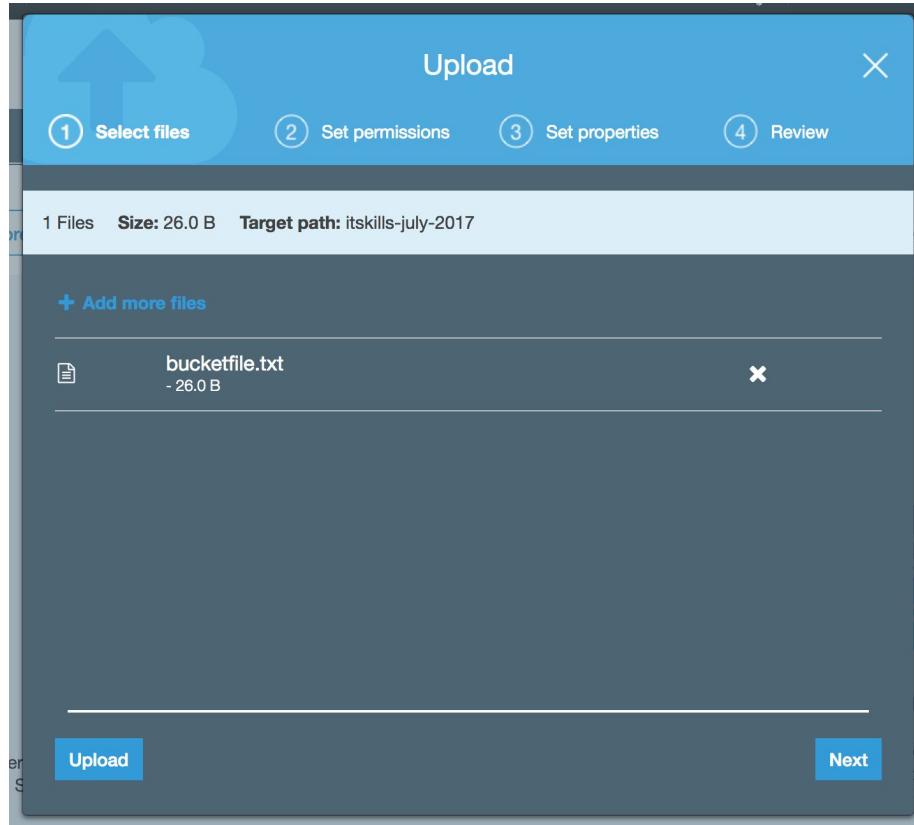
[Learn more](#)

 Disabled

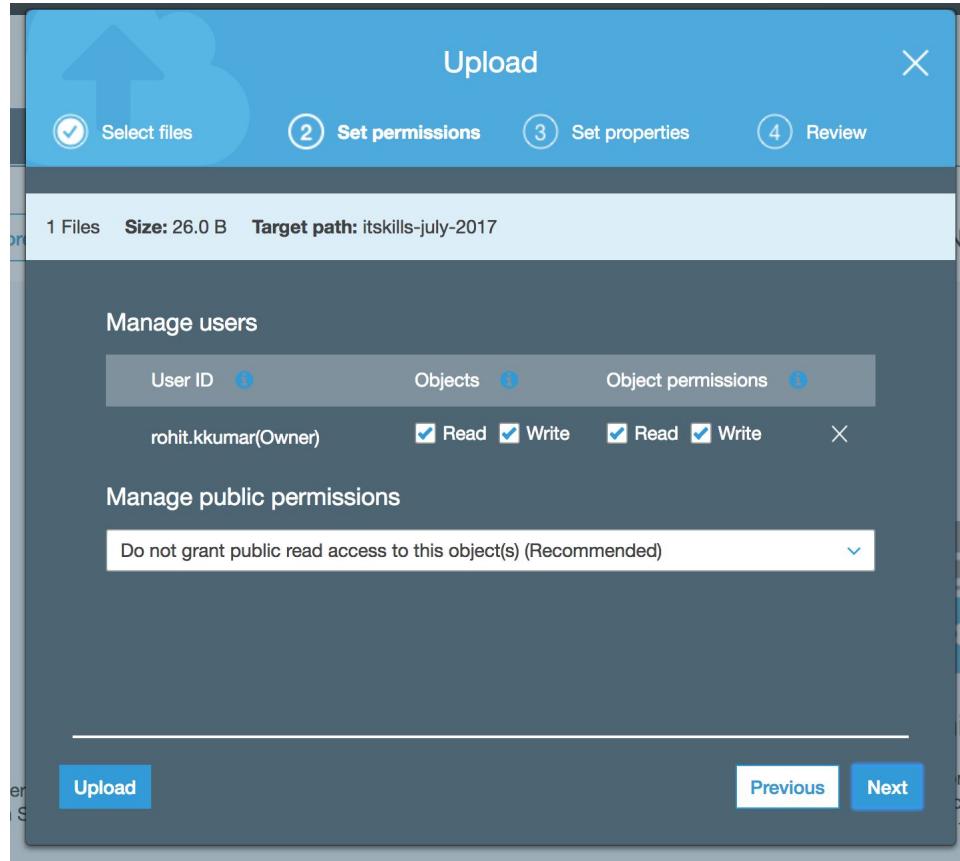
Upload File - 1



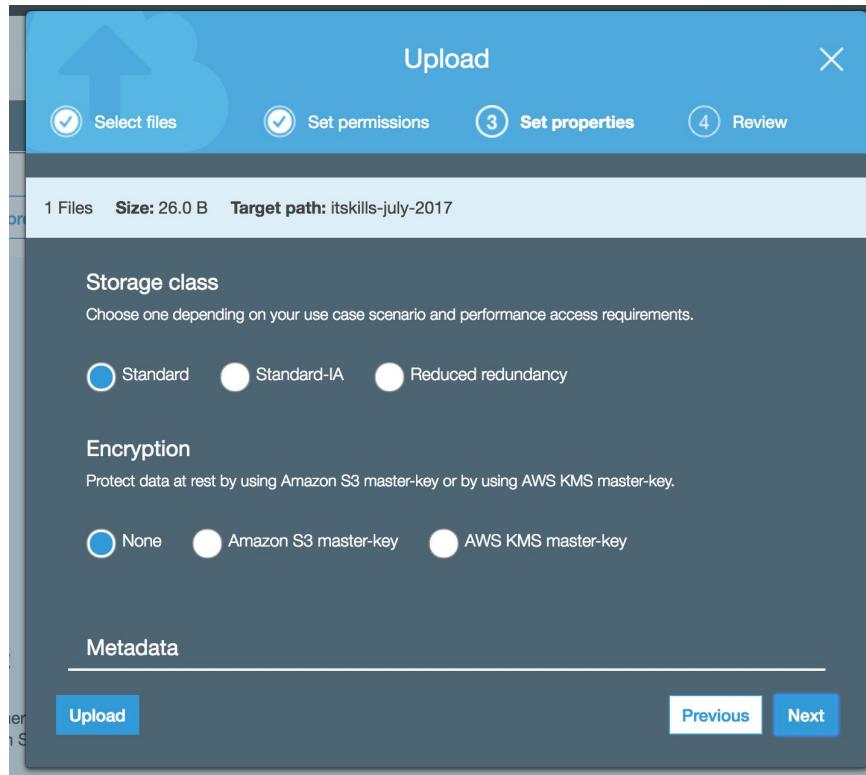
Upload File - 2



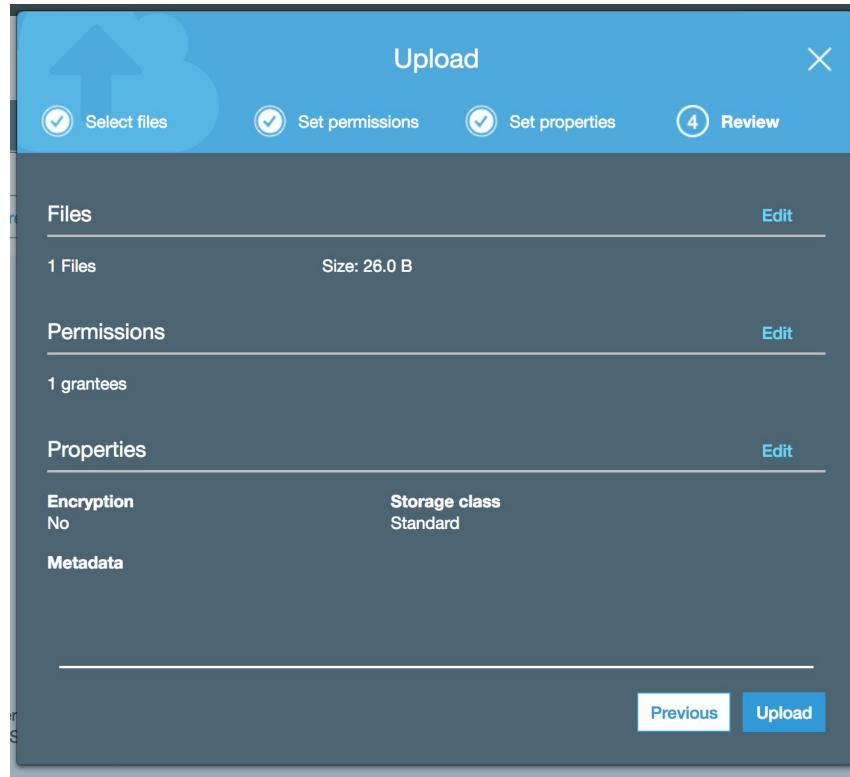
Set File Permissions



File Properties



Review & Upload





Amazon S3 > itskills-july-2017

Overview

Properties

Permissions

Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload

+ Create folder

More ▾

US East (N. Virginia)

Viewing 1 to 1

Name ↑↓

Last modified ↑↓

Size ↑↓

Storage class ↑↓

bucketfile.txt

Jul 22, 2017 1:27:57 AM

26.0 B

Standard

Viewing 1 to 1

File Properties



Owner

rohit.kkumar

Last modified

Jul 22, 2017 1:27:57 AM

Etag

0877c966aa49227f959c9d0083bee379

Storage class

Standard

Server side encryption

None

Size

26

Link

<https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt>

File Version

bucketfile.txt [Latest version ▾](#)



Jul 22, 2017 1:27:57 AM (Latest version)

Standard



File Access

← → C 🔒 Secure | <https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt>

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>C7CAC8F4E8C0B705</RequestId>
  ▼<HostId>
    j4iN1jXYyebMkio2uDxhD9CLfWUV/eboOq0rSuE02YdgKAravEi0lq/BDXgsBmxjSGLRQVPoPMY=
  </HostId>
</Error>
function(){ window.SIG_EXT = {}; } )()
```

[Open](#)

[Download](#)

[Download as](#)

[Make public](#)

[Copy path](#)



Secure

<https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt>

this is the first version

Key Things

- Owner
- Owner Permissions
- Permissions
- Unique Link

Practical: Create a Versioned File

Enable versioning

The screenshot shows the Amazon S3 console interface. At the top, there is a navigation bar with the Amazon S3 logo and the bucket name "itskills-july-2017". Below the navigation bar is a horizontal menu with four tabs: "Overview" (selected), "Properties", "Permissions", and "Management". A modal dialog box titled "Versioning" is displayed in the center. The dialog contains two radio button options: "Enable versioning" (selected) and "Suspend versioning". A descriptive text below the options states: "This suspends the creation of object versions for all operations but preserves any existing object versions." At the bottom of the dialog are two buttons: "Cancel" and "Save" (highlighted with a blue border).

Amazon S3 > itskills-july-2017

Overview Properties Permissions Management

Versioning

Enable versioning

Suspend versioning

This suspends the creation of object versions for all operations but preserves any existing object versions.

Cancel Save

Versioning enabled

Amazon S3 > itskills-july-2017

Overview **Properties** **Permissions** **Management**

Versioning

Keep multiple versions of an object in the same bucket.

[Learn more](#)

Enabled

Logging

Set up access log records tha details about access requ

[Learn more](#)

Disabled

Upload updated file

The screenshot shows the Amazon S3 console interface. At the top left is the AWS logo. Next to it, the path 'Amazon S3 > itskills-july-2017' is displayed. Below this, the file name 'bucketfile.txt' is shown, followed by a link 'Latest version ▾'. The main area displays two versions of the file:

Last Modified	Storage Class	Actions
Jul 22, 2017 2:06:56 AM (Latest version)	Standard	
Jul 22, 2017 1:27:57 AM	Standard	

Each row has a small blue bar at the bottom.

It Works !!!

← → C  Secure | <https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt>

this is the first version
this is the second version

Key Things

- ETag
- Size
- Versions
- Careful about the size of file during architecting.
- Restore a deleted file

Versioning

- Keep multiple versions of an object in one bucket
- Helps protect your data against accidental or malicious deletion by keeping multiple versions of each object in the bucket
- Preserve, retrieve, and restore every version of every object
- Turned on at the bucket level.
- Versioning cannot be removed from a bucket; it can only be suspended
- MFA Delete adds another layer of data protection on top of bucket versioning.

Practical: Restore Deleted Object

Practical: Cross Region Replication

Source Bucket Properties

Advanced settings

Tags

Use tags to track your cost against projects or other criteria.

[Learn more](#)

 0 Tags

Cross-region replication

Automate copying objects across different AWS Regions.

[Learn more](#)

 Disabled

Transfer acceleration

Enable fast, easy and secure transfers of files to and from your bucket.

[Learn more](#)

 Suspended

Events

Receive notifications when specific events occur in your bucket.

[Learn more](#)

 0 Active notifications

Requester pays

The requester (instead of the bucket owner) will pay for requests and data transfer.

[Learn more](#)

 Disabled

Enable Replication

Cross-region replication

Enable cross-region replication i

To use cross-region replication, you must enable versioning on this bucket and on the target bucket.

Disable cross-region replication

Delete configuration

Cancel **Save**

Fill in Replica Details

Cross-region replication

Enable cross-region replication i

Source

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

Select a source

Destination

Select a region

Select bucket

Select role

Select an IAM role

Disable cross-region replication

Cancel Save

Existing objects will not be replicated. Cross-region replication replicates all future uploads of every object to another bucket.

Cross-region replication

Enable cross-region replication i

Source

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

Whole bucket

Destination

Asia Pacific (Mumbai)

itskills-july-2017-replica

Destination storage class

Same as source object (Default)

Select role

Create new role

Disable cross-region replication

Cancel Save

Replication ON

Advanced settings

Tags

Use tags to track your cost against projects or other criteria.

[Learn more](#)



0 Tags

Cross-region replication

Automate copying objects across different AWS Regions.

[Learn more](#)



Destination bucket:
itskills-july-2017-replica

Old object not replicated

The screenshot shows the Amazon S3 console interface for a bucket named "itskills-july-2017-replica". The top navigation bar includes the S3 logo, the bucket name, and links for Overview, Properties, Permissions, and Management. Below the navigation is a toolbar with Upload, Create folder, More, All (selected), and Deleted objects buttons. The main content area displays a message: "This bucket is empty. Upload new objects to get started." The region is set to Asia Pacific (Mumbai) with a refresh icon.

Amazon S3 > itskills-july-2017-replica

Overview Properties Permissions Management

Upload Create folder More All Deleted objects

Asia Pacific (Mumbai)

This bucket is empty. Upload new objects to get started.

Object Replicated

The screenshot shows the Amazon S3 console interface. At the top, there is a navigation bar with the Amazon S3 logo and the path "Amazon S3 > itskills-july-2017-replica". Below the navigation bar, there is a tab menu with four options: "Overview", "Properties" (which is highlighted in blue), "Permissions", and "Management".

Below the tab menu is a search bar with the placeholder text "Type a prefix and press Enter to search. Press ESC to clear." followed by a magnifying glass icon.

On the left side of the main content area, there are several buttons: "Upload", "+ Create folder", "More ▾", and two tabs: "All" (which is highlighted in blue) and "Deleted objects". On the right side, it shows the region "Asia Pacific (Mumbai)" and a refresh icon.

The main content area displays a table of objects. The columns are: "Name" (with a checkbox header), "Last modified", "Size", and "Storage class". There is one object listed:

Name	Last modified	Size	Storage class
bucketfile.txt	Jul 22, 2017 2:15:52 AM	79.0 B	Standard

At the bottom of the page, there is a footer bar with the text "Viewing 1 to 1" and navigation arrows.

Replicated Source File

Overview Properties Permissions

Open Download Download as Make public Copy path

Owner
rohit.kkumar

Last modified
Jul 22, 2017 2:15:52 AM

Etag
a9386f4a6e912c17cd9e34541bff4cad

Storage class
Standard

Server side encryption
None

Replication status
COMPLETED

Size
79

Version ID
zieZ8fnYegU6oxKUfvxnWJum4gQxALGc

Link
<https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt>

Delete the file

The screenshot shows the Amazon S3 console interface. At the top, there is a navigation bar with a red icon, the text "Amazon S3 > itskills-july-2017", and four tabs: "Overview", "Properties" (which is selected), "Permissions", and "Management". Below the navigation bar is a search bar with the placeholder text "Type a prefix and press Enter to search. Press ESC to clear." To the right of the search bar are two buttons: "Upload" and "+ Create folder". Further down are buttons for "More", "All" (selected), and "Deleted objects". A sidebar on the left contains a checkbox for "Name" and a list item for "bucketfile.txt" which also has a selected checkbox. A context menu is open over the "bucketfile.txt" item, with options: "Get size", "Download as", "Rename", "Delete" (which is highlighted in blue), "Undo delete", "Cut", and "Copy". To the right of the context menu is a detailed view of the file "bucketfile.txt". The file name is "bucketfile.txt". The "Latest version" is shown. The "Overview" section provides the following details:

Key	bucketfile.txt
Size	79
Expiration date	N/A
Expiration rule	N/A
ETag	a9386f4a6e912c17cd9e34541bff4cad
Last modified	Jul 22, 2017 2:31:15 AM GMT+0530
Link	https://s3.amazonaws.com/itskills-july-2017/bucketfile.txt

Delete Marker

The screenshot shows the Amazon S3 console interface for a bucket named "itskills-july-2017". The top navigation bar includes the S3 logo, the bucket name, and tabs for Overview, Properties (selected), Permissions, and Management. A search bar at the top allows for prefix searching. Below the search bar are buttons for Upload, Create folder, and More, followed by a dropdown menu set to "Deleted objects". The main content area displays a list of deleted objects. One object, "bucketfile.txt", is highlighted. A modal window for "bucketfile.txt" is open, showing its properties. The modal has a close button (X) in the top right. Inside, the file name "bucketfile.txt" is displayed above two buttons: "Download" and "Copy path". A dark grey footer bar in the modal contains the text "Latest version ▾". The main list below shows "bucketfile.txt" with a checkmark next to it, indicating it is selected. The list includes columns for Name, Last modified, Size, and Actions (download and delete icons). The "Last modified" column shows "Jul 22, 2017 2:31:15 AM (Latest version)" and "Jul 22, 2017 2:32". The "Size" column shows "-". The "Actions" column shows download and delete icons. The "Expiration date" and "Expiration rule" fields are listed under the "Properties" section of the modal.

bucketfile.txt

Download Copy path

Latest version ▾

Name Jul 22, 2017 2:31:15 AM (Latest version) Standard

bucketfile.txt Jul 22, 2017 2:32 -

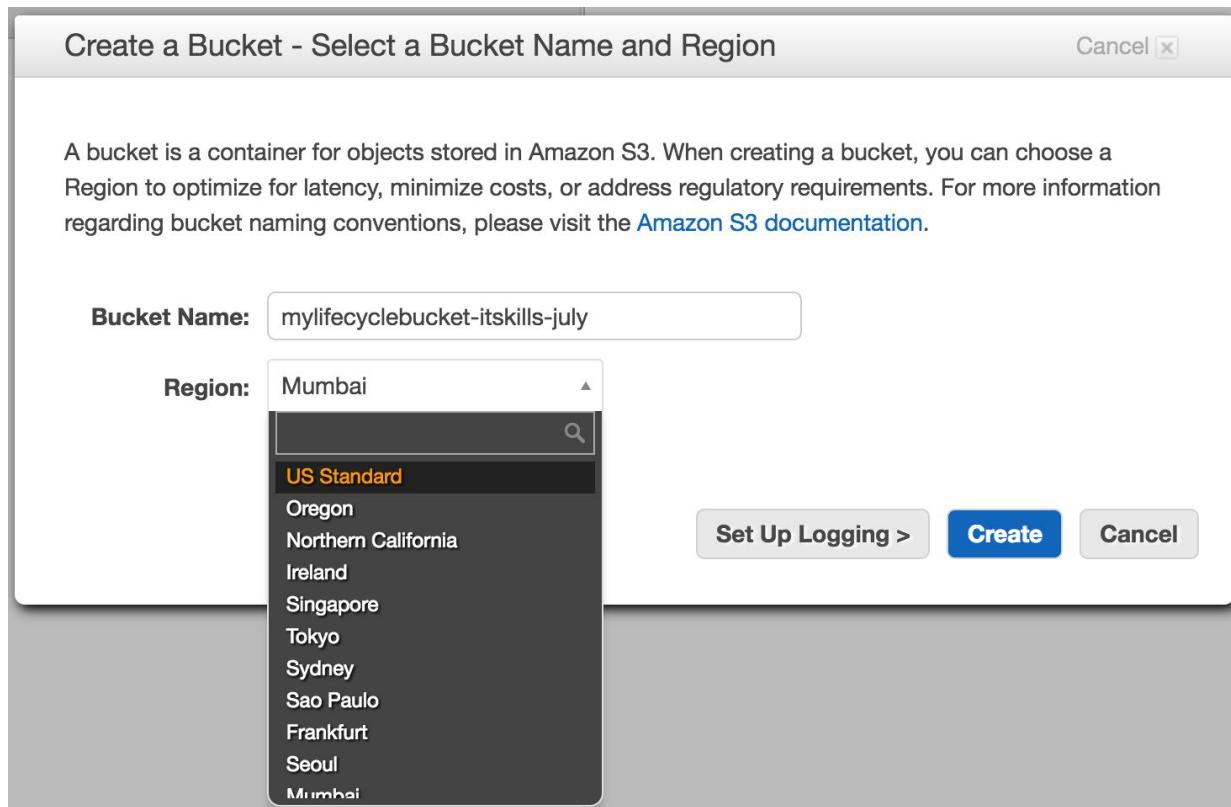
Size -
Expiration date
Expiration rule
ETag -
Last modified Jul 22, 2017 2:32:04 AM GMT+0530

Learnings

- Versioning must be enabled on both the buckets
- Regions must be unique, CORS will not work on buckets in same region
- Files in existing bucket are not replicated automatically
- All subsequent files will be replicated automatically
- You cannot replicate multiple buckets / chain buckets
- Delete markers are replicated
- Deleting individual versions / deleting the marker files do not get replicated

Practical: Lifecycle management & Glacier

Create a new bucket



This screenshot shows the AWS Regions selection interface. It includes a header with 'Rohit Kumar' and 'N. Virginia'. A sidebar on the left lists regions under 'US East (N. Virginia)'. The main area lists regions grouped by continent: US, Asia Pacific, and South America.

Region
US East (Ohio)
US West (N. California)
US West (Oregon)
Canada (Central)
EU (Ireland)
EU (Frankfurt)
EU (London)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Seoul)
Asia Pacific (Tokyo)
Asia Pacific (Mumbai)
South America (São Paulo)

G
L
A
C
I
E
R
R
E
G
I
O
N
S

Old Console / Lifecycle

▼ Lifecycle

You can manage the lifecycle of objects by using [Lifecycle rules](#). Lifecycle rules enable you to automatically transition objects to the [Standard - Infrequent Access](#) Storage Class, and/or archive objects to the [Glacier](#) Storage Class, and/or remove objects after a specified time period. Rules are applied to all the objects that share the specified prefix.

Versioning is not currently enabled on this bucket.

You can use Lifecycle rules to manage all versions of your objects. This includes both the Current version and Previous versions.

 **Add rule**

Save

Cancel

Select bucket or partial

Lifecycle Rules

Step 1: Choose Rule Target

Step 2: Configure Rule

Step 3: Review and Name

Apply the Rule to:

Whole Bucket: mylifecyclebucket-itskills-july

A Prefix e.g. MyFolder/ or MyFolder/MyObject

Lifecycle Rule

Lifecycle rules will help you manage your storage costs by controlling the lifecycle of your objects. Create Lifecycle rules to automatically transition your objects to the Standard - Infrequent Access Storage Class, archive them to the Glacier Storage Class, and remove them after a specified time period.

Choose different options below to see what works best for your use case. No rule will take effect until you activate them at the end of this wizard.

Action on Objects

- Transition to the Standard - Infrequent Access Storage Class** Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

- Archive to the Glacier Storage Class** Days after the object's creation date

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are [not immediately accessible](#).

- Permanently Delete** Days after the object's creation date

Lifecycle Rule

Action on Objects

- Transition to the Standard - Infrequent Access Storage Class

30

Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

- Archive to the Glacier Storage Class

Days after the object's creation date

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

- Permanently Delete

Days after the object's creation date

EXAMPLE:



Lifecycle Rule Name

Rule Name

Choose a descriptive name for your rule so you can easily identify it in the future. If you do not want to enter a name now, we will generate one for you.

Rule Name: (Optional)

Rule Target

[Edit](#)

This rule will apply to the whole bucket: **mylifecyclebucket-itskills-july**

Rule Configuration

[Edit](#)

Action on Objects

Transition to the Standard - Infrequent Access Storage Class **30** days after the object's creation date.

Archive to the Glacier Storage Class **60** days after the object's creation date.

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

Permanently Delete **425** days after the object's creation date

As versioning is not enabled, lifecycle delete rule will permanently delete the objects with no recovery.

[Cancel](#)

[**< Configure Rule**](#)

Save Rule

Practical: Lifecycle management & Glacier + Versioning

Create a new bucket

Create a Bucket - Select a Bucket Name and Region Cancel

A bucket is a container for objects stored in Amazon S3. When creating a bucket, you can choose a Region to optimize for latency, minimize costs, or address regulatory requirements. For more information regarding bucket naming conventions, please visit the [Amazon S3 documentation](#).

Bucket Name: mylifecyclebucket-itskills-july-versioned

Region: Select a Region

- US Standard
- Oregon
- Northern California
- Ireland
- Singapore
- Tokyo
- Sydney
- Sao Paulo
- Frankfurt
- Seoul
- Mumbai

Set Up Logging > Create Cancel

Enable Versioning on the bucket

▼ Versioning

[Versioning](#) allows you to preserve, retrieve, and restore every version of every object stored in this bucket. This provides an additional level of protection by providing a means of recovery for accidental overwrites or expirations. Versioning-enabled buckets store all versions of your objects by default.

You can use Lifecycle rules to manage all versions of your objects as well as their associated costs. Lifecycle rules enable you to automatically archive your objects to the Glacier Storage Class and/or remove them after a specified time period.

Once enabled, Versioning cannot be disabled, only suspended.

Versioning is currently enabled on this bucket.

[Suspend Versioning](#)

Add a lifecycle Rule

▼ Lifecycle

You can manage the lifecycle of objects by using [Lifecycle rules](#). Lifecycle rules enable you to automatically transition objects to the [Standard - Infrequent Access](#) Storage Class, and/or archive objects to the [Glacier](#) Storage Class, and/or remove objects after a specified time period. Rules are applied to all the objects that share the specified prefix.

Versioning is currently enabled on this bucket.

You can use Lifecycle rules to manage all versions of your objects. This includes both the Current version and Previous versions.

 **Add rule**

Save

Cancel

Lifecycle Rule

Action on Current Version

- Transition to the Standard - Infrequent Access Storage Class

Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

- Archive to the Glacier Storage Class

Days after the object's creation date

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

- Expire

Days after the object's creation date

For versioning-enabled buckets, an expire will retain the current version as a previous version and place a delete marker as the current version. If you wish to permanently delete previous versions, combine the **Expire** action here with the **Permanently Delete** previous versions action below.

Action on Previous Versions

- Transition to the Standard - Infrequent Access Storage Class

Days after becoming a previous version

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

- Archive to the Glacier Storage Class

Days after becoming a previous version

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

- Permanently Delete

Days after becoming a previous version

This rule will permanently delete a previous version of an object as the version becomes eligible for expiration. You cannot recover permanently deleted versions of objects.

Cancel

< Set Target

Review >

Action on Current Version

Transition to the Standard - Infrequent Access Storage Class

29



Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

A minimum of 30 days is required before transitioning to the
Standard - Infrequent Access Storage Class

!

Please enter an integer value greater than or equal to 30.

Action on Current Version

Transition to the Standard - Infrequent Access Storage Class

30



Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

Archive to the Glacier Storage Class

59

Days after the object's creation date

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.



An object must remain in the Standard - Infrequent Access Storage Class for a minimum of 30 days before transitioning to the Glacier Storage Class

Please enter an integer value greater than or equal to 60.

Action on Current Version

- Transition to the Standard - Infrequent Access Storage Class

30

Days after the object's creation date

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

- Archive to the Glacier Storage Class

60

Days after the object's creation date

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

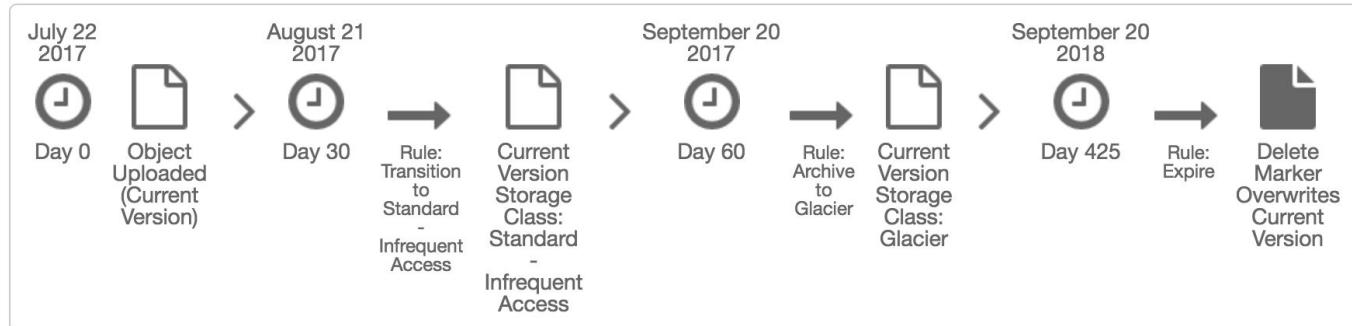
- Expire

425

Days after the object's creation date

For versioning-enabled buckets, an expire will retain the current version as a previous version and place a delete marker as the current version. If you wish to permanently delete previous versions, combine the **Expire** action here with the **Permanently Delete** previous versions action below.

EXAMPLE:



Action on Previous Versions

Transition to the Standard - Infrequent Access Storage Class

30 Days after becoming a previous version

Standard - Infrequent Access has a 30-day minimum retention period and a 128KB minimum object size. Lifecycle policy will not transition objects that are less than 128KB. Refer [here](#) to learn more about Standard - Infrequent Access.

Archive to the Glacier Storage Class

60 Days after becoming a previous version

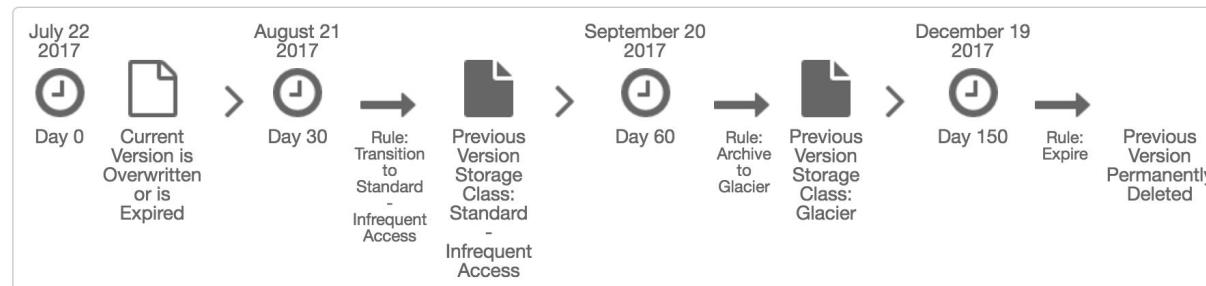
This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are **not immediately accessible**.

Permanently Delete

150 Days after becoming a previous version

This rule will permanently delete a previous version of an object as the version becomes eligible for expiration. You cannot recover permanently deleted versions of objects.

EXAMPLE:



Cancel

< Set Target

Review >

Save the rule

Rule Name

Choose a descriptive name for your rule so you can easily identify it in the future. If you do not want to enter a name now, we will generate one for you.

Rule Name: (Optional)

Rule Target

Edit

This rule will apply to the whole bucket: **mylifecyclebucket-itskills-july-versioned**

Rule Configuration

Edit

Action on Current Version

Transition to the Standard - Infrequent Access Storage Class **30** days after the object's creation date.

Archive to the Glacier Storage Class **60** days after the object's creation date.

This rule could reduce your storage costs. Refer [here](#) to learn more about Glacier pricing. Note that objects archived to the Glacier Storage Class are [not immediately accessible](#).

Expire **425** days after the object's creation date

For versioning-enabled buckets, an expire will retain the current version as a previous version and place a delete marker as the current version. If you wish to permanently delete previous versions, combine the **Expire** action here with the **Permanently Delete** previous versions action below.

Action on Previous Versions

Cancel

< Configure Rule

Create and Activate Rule

Pricing for N. Virginia / Could vary with region

For Requests Not Otherwise Specified Below

PUT, COPY, POST, or LIST Requests	\$0.005 per 1,000 requests
-----------------------------------	----------------------------

GET and all other Requests	\$0.004 per 10,000 requests
----------------------------	-----------------------------

Delete Requests	Free †
-----------------	--------

For Standard – Infrequent Access Requests

PUT, COPY, or POST Requests	\$0.01 per 1,000 requests
-----------------------------	---------------------------

GET and all other Requests	\$0.01 per 10,000 requests
----------------------------	----------------------------

Lifecycle Transition Requests into Standard – Infrequent Access	\$0.01 per 1,000 requests
---	---------------------------

Data Retrievals	\$0.01 per GB
-----------------	---------------

For Glacier Requests

Lifecycle Transition Requests into Glacier	\$0.05 per 1,000 requests
--	---------------------------

Glacier Retrieval Fees	See Glacier Pricing Page
------------------------	--

Practical: Static Website Hosting

Static Website Hosting

Static websites have many advantages: they are very fast, very scalable, and can be more secure than a typical dynamic website. If you host a static website on Amazon S3, you can also leverage the security, durability, availability, and scalability of Amazon S3.

Because every Amazon S3 object has a URL, it is relatively straightforward to turn a bucket into a website. To host a static website, you simply configure a bucket for website hosting and then upload the content of the static website to the bucket. To configure an Amazon S3 bucket for static website hosting:

Static Website Hosting

- Before You Begin
- Step 1: Register a Domain
- Step 2: Create and Configure Buckets and Upload Data
- Step 3: Create and Configure Amazon Route 53 Hosted Zone
- Step 4: Switch to Amazon Route 53 as Your DNS Provider
- Step 5: Testing

1. Create a bucket with the same name as the desired website hostname.
2. Upload the static files to the bucket.
3. Make all the files public (world readable).
4. Enable static website hosting for the bucket. This includes specifying an Index document and an Error document.
5. The website will now be available at the S3 website URL:
`<bucket-name>.s3-website-<AWS-region>.amazonaws.com`.
6. Create a friendly DNS name in your own domain for the website using a DNS CNAME, or an Amazon Route 53 alias that resolves to the Amazon S3 website URL.
7. The website will now be available at your website domain name.

Glacier

Amazon Glacier

Amazon Glacier is an **extremely low-cost storage service** that provides durable, secure, and flexible **storage for data archiving and online backup**. To keep costs low, Amazon Glacier is **designed for infrequently accessed data where a retrieval time of three to five hours is acceptable**. Amazon Glacier can **store an unlimited amount of virtually any kind of data, in any format**.

Common use cases for Amazon Glacier include replacement of traditional tape solutions for long-term backup and archive and storage of data required for compliance purposes.

In most cases, the data stored in Amazon Glacier **consists of large TAR (Tape Archive) or ZIP files**.

Like Amazon S3, Amazon Glacier is extremely durable, storing data on multiple devices across multiple facilities in a region. Amazon Glacier is designed for **99.99999999% durability** of objects over a given year.

Vaults

Vaults are **containers for archives**. Each AWS account can have up to 1,000 **vaults**. You can control access to your vaults and the actions allowed using IAM policies or vault access policies.

Vaults Locks

You can easily **deploy and enforce compliance controls for individual Amazon Glacier vaults with a vault lock policy.** You can specify controls such as Write Once Read Many (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

Data Retrieval

You can retrieve up to **5% of your data stored in Amazon Glacier for free each month**, calculated on a daily prorated basis. If you retrieve more than 5%, you will incur retrieval fees based on your maximum retrieval rate. To eliminate or minimize those fees, you can set a data retrieval policy on a vault to limit your retrievals to the free tier or to a specified data rate.

Conclusion

Best Practices, Patterns, and Performance

- Data in on-premises file systems, databases, and compliance archives can easily be backed up over the Internet to Amazon S3 or Amazon Glacier, while the primary application or database storage remains on-premises
- S3 will scale automatically to support very high request rates, automatically repartitioning your buckets as needed
- Glacier, data is stored in archives. An archive can contain up to 40TB of data, and you can have an unlimited number of archives. Each archive is assigned a unique archive ID at the time of creation.
- All archives are automatically encrypted, and archives are immutable—after an archive is created, it cannot be modified
- You can retrieve up to 5% of your data stored in Amazon Glacier for free each month
- Amazon S3 standard storage is designed for 11 nines (99.9999999%) durability and four nines (99.99%) availability of objects over a year

Exam Essentials

- Know what amazon s3 is and what it is commonly used for
- Understand how object storage differs from block and file storage.
- Understand the basics of Amazon S3.
- Understand the durability, availability, and data consistency model of Amazon S3.
- Know how to enable static website hosting on Amazon S3.
- Know how to protect your data on Amazon S3.
- Know the use case for each of the Amazon S3 storage classes.
- Know how to use lifecycle configuration rules
- Know how to use Amazon S3 event notifications
- Know the basics of amazon glacier as a standalone service

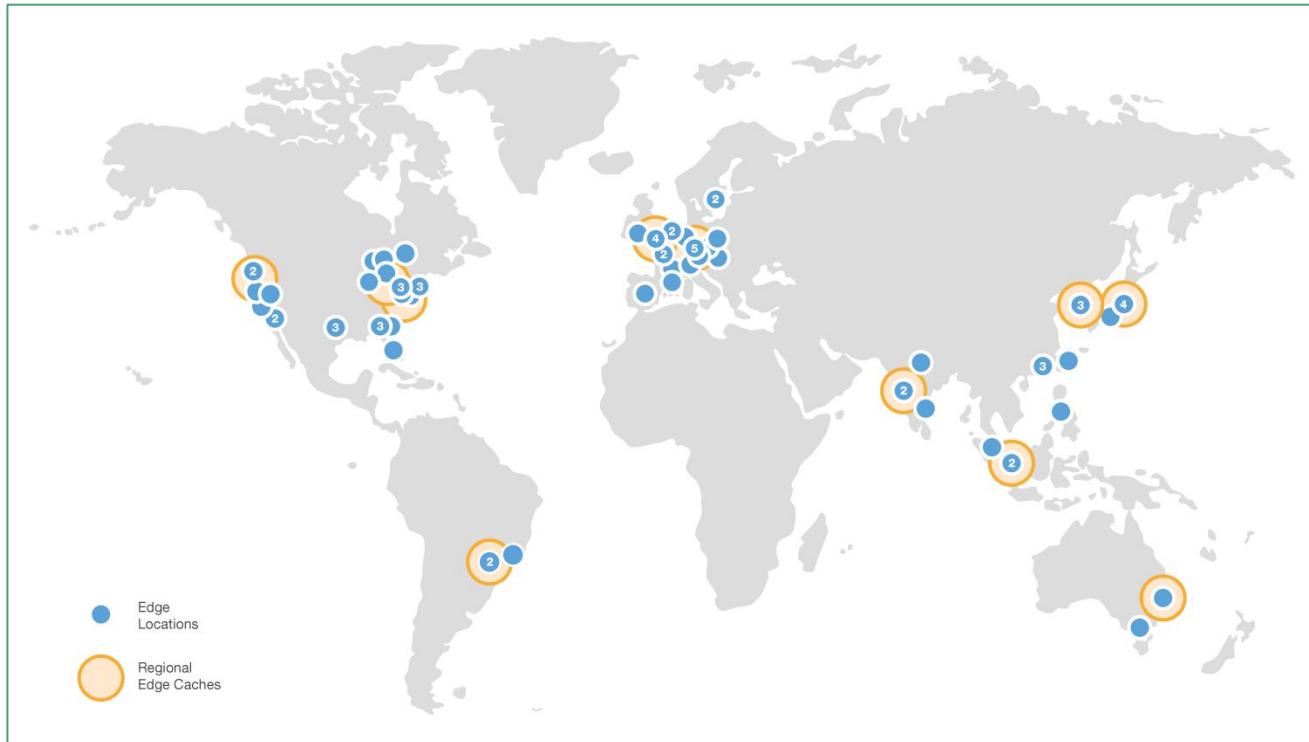
CDN

Amazon CloudFront - CDN

Amazon CloudFront is a **web service that speeds up distribution of your static and dynamic web content**, for example, .html, .css, .php, image, and media files, to end users. CloudFront delivers your content through a **worldwide network of edge locations**. When an end user requests content that you're serving with CloudFront, the **user is routed to the edge location that provides the lowest latency**, so content is delivered with the best possible performance. If the content is already in that edge location, CloudFront delivers it immediately. If the content is not currently in that edge location, **CloudFront retrieves it from an Amazon S3 bucket or an HTTP server** (for example, a web server) that you have identified as the source for the definitive version of your content.

The Amazon CloudFront Global Edge Network

To deliver content to end users with lower latency, Amazon CloudFront uses a global network of 79 edge locations and 11 regional edge caches across 22 countries and 49 cities for content delivery.



Viewing an Image / Traceroute



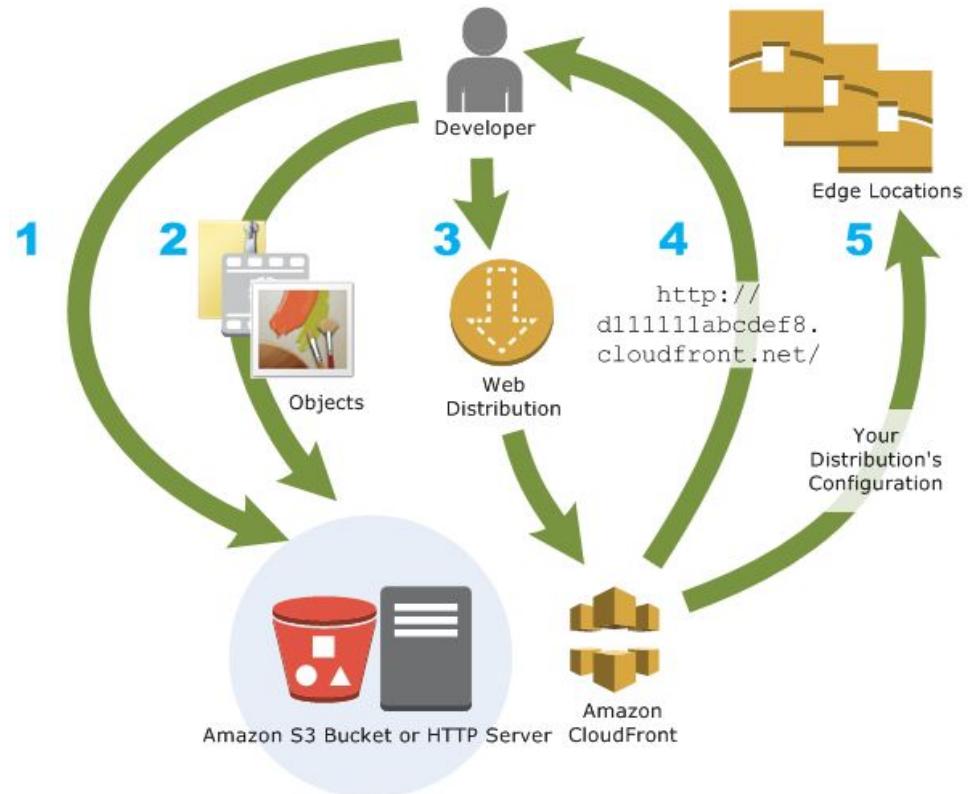
1	vrid-225.core-sw.aus.us.siteprotect.com (216.139.225.1) 0.627 ms
2	xe-3-4.brdr-rtr-02.aus.us.siteprotect.com (216.139.253.53) 0.219 ms
3	66.113.197.121 0.452 ms
4	xe-5-2-0.edge3.Dallas1.Level3.net (4.59.112.37) 4.978 ms
5	ae-73-70.ebr3.Dallas1.Level3.net (4.69.145.116) 9.817 ms
6	ae-7-7.ebr3.Atlanta2.Level3.net (4.69.134.22) 30.570 ms
7	ae-2-2.ebr1.Washington1.Level3.net (4.69.132.86) 38.801 ms
8	ae-81-81.csv3.Washington1.Level3.net (4.69.134.138) 41.795 ms
9	ae-3-89.edge2.Washington1.Level3.net (4.68.17.145) 39.193 ms
10	72.21.222.139 35.767 ms

The request was routed 10 times within the United States before the image was retrieved

How CloudFront Delivers Content

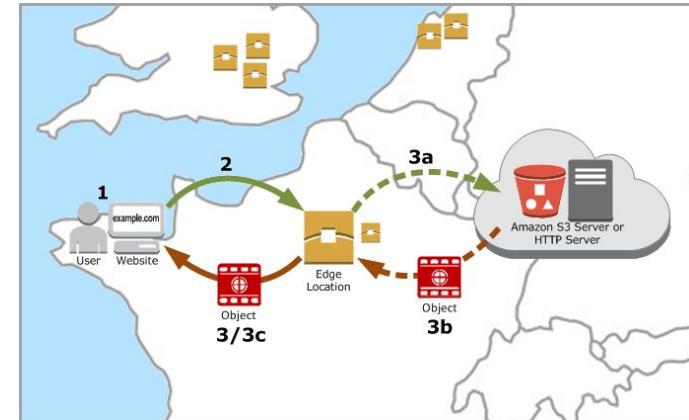
How You Configure CloudFront

- You configure your *origin servers*, from which CloudFront gets your files.
- You upload your files to your origin servers.
- You create a CloudFront *distribution*, which tells CloudFront which origin servers to get your files from.
- CloudFront assigns a domain name to your new distribution and displays it in the CloudFront console.
- CloudFront sends your distribution's configuration (but not your content) to all of its **edge locations**, where CloudFront caches copies of your objects.



How CloudFront Delivers Content to Your Users

- A user requests one or more objects, such as an image file and an HTML file.
- DNS routes the request to the CloudFront edge location that can best serve the user's request, typically the nearest CloudFront edge location in terms of latency, and routes the request to that edge location.
- In the edge location, CloudFront checks its cache for the requested files. If the files are in the cache, CloudFront returns them to the user. If the files are *not* in the cache, it does the following:
 - a. CloudFront compares the request with the specifications in your distribution and forwards the request for the files to the applicable origin server for the corresponding file type—for example, to your Amazon S3 bucket for image files and to your HTTP server for the HTML files.
 - b. The origin servers send the files back to the CloudFront edge location.
 - c. As soon as the first byte arrives from the origin, CloudFront begins to forward the files to the user. CloudFront also adds the files to the cache in the edge location for the next time someone requests those files.



What is CloudFront?

Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming and interactive content using a global network of edge locations.

Requests for your content are automatically routed to your nearest edge location, so content is delivered with the best possible performance.

Optimized to work with S3, EC2 or non Amazon service which stores the original definitive version of your files.

CloudFront Key Terms

- Edge Location - This is the place where **content is cached**. This is different from Region / AZ
- Origin - This is the origin of all the files that the CDN will distribute. This can be either **S3 bucket / EC2 instance** / an ELB or Route 53 / Non Amazon Custom Origin
- Distribution - Collection of Edge Locations
 - Web Distributions
 - RTMP (Media) distributions

Tips

- Edge locations are where content is cached.
- Over 50+ edge locations
- Edge locations are not read only. You can PUT / write to edge locations too
- Objects are cached for TTL
- You can manually clear the cache before the TTL, but you will be charged for that

Practical: Creating a CloudFront
