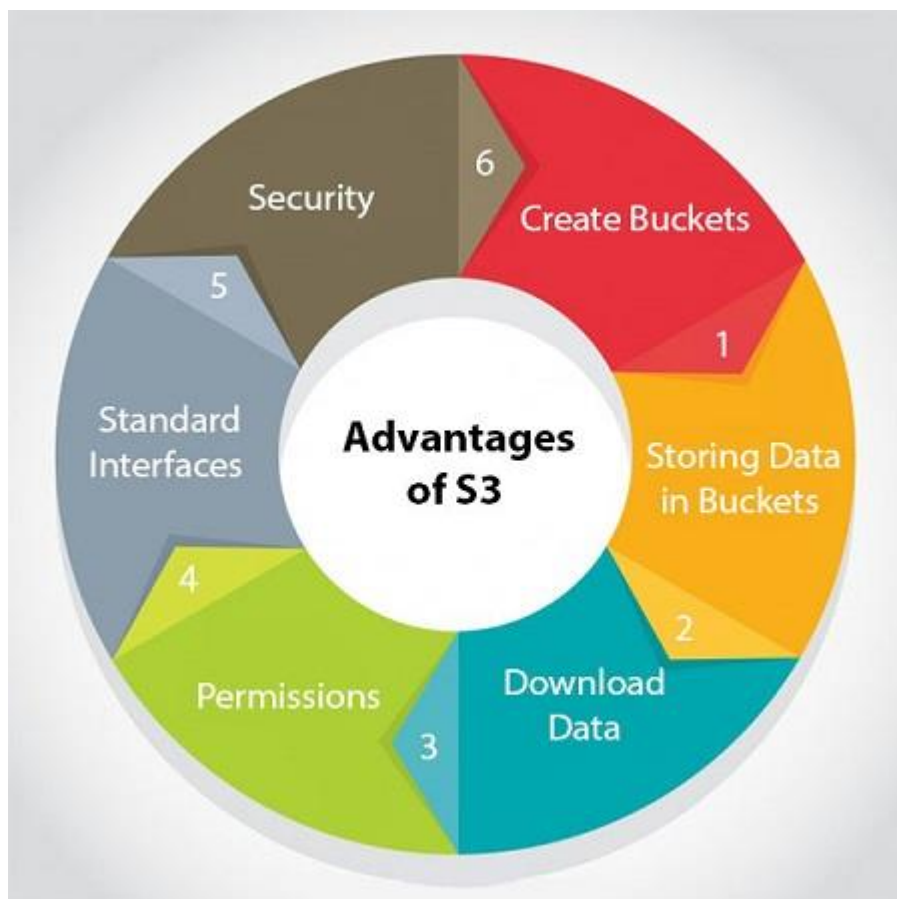


What is S3?

- S3 is a safe place to store the files.
- It is Object-based storage, i.e., you can store the images, word files, pdf files, etc.
- The files which are stored in S3 can be from 0 Bytes to 5 TB.
- It has unlimited storage means that you can store the data as much you want.
- Files are stored in Bucket. A bucket is like a folder available in S3 that stores the files.
- S3 is a universal namespace, i.e., the names must be unique globally. Bucket contains a DNS address. Therefore, the bucket must contain a unique name to generate a unique DNS address.
- If you upload a file to S3 bucket, then you will receive an HTTP 200 code means that the uploading of a file is successful.

Advantages of Amazon S3



- **Create Buckets:** Firstly, we create a bucket and provide a name to the bucket. Buckets are the containers in S3 that stores the data. Buckets must have a unique name to generate a unique DNS address.
- **Storing data in buckets:** Bucket can be used to store an infinite amount of data. You can upload the files as much you want into an Amazon S3 bucket, i.e., there is no maximum limit to store the files. Each object can contain upto 5 TB of data. Each object can be stored and retrieved by using a unique developer assigned-key.
- **Download data:** You can also download your data from a bucket and can also give permission to others to download the same data. You can download the data at any time whenever you want.
- **Permissions:** You can also grant or deny access to others who want to download or upload the data from your Amazon S3 bucket. Authentication mechanism keeps the data secure from unauthorized access.
- **Standard interfaces:** S3 is used with the standard interfaces REST and SOAP interfaces which are designed in such a way that they can work with any development toolkit.
- **Security:** Amazon S3 offers security features by protecting unauthorized users from accessing your data.

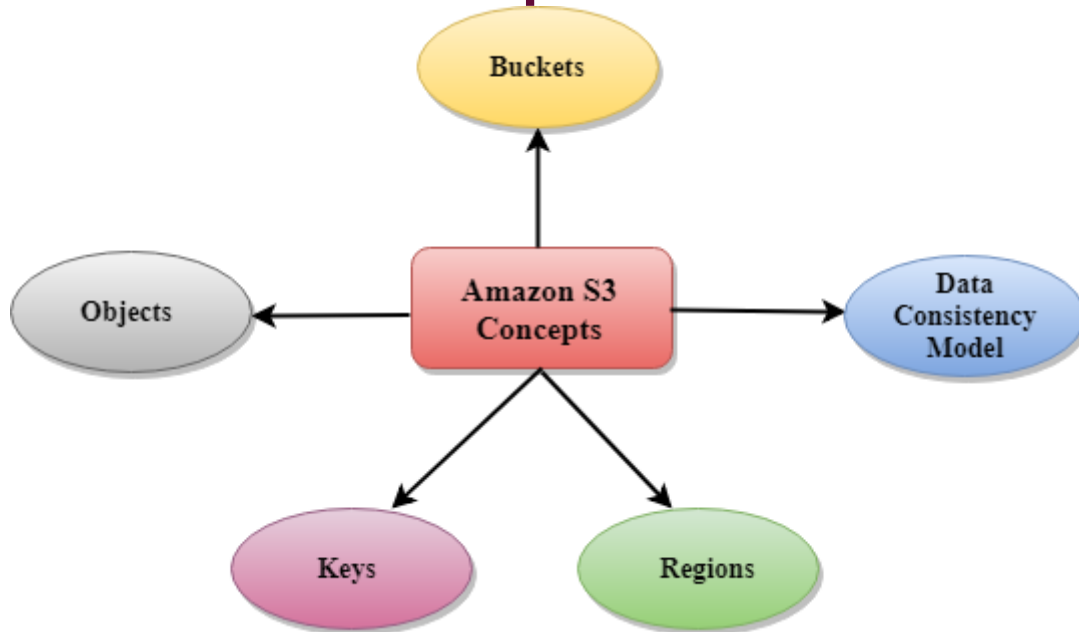
S3 is a simple key-value store

S3 is object-based. Objects consist of the following:

- **Key:** It is simply the name of the object. For example, hello.txt, spreadsheet.xlsx, etc. You can use the key to retrieve the object.
- **Value:** It is simply the data which is made up of a sequence of bytes. It is actually a data inside the file.
- **Version ID:** Version ID uniquely identifies the object. It is a string generated by S3 when you add an object to the S3 bucket.
- **Metadata:** It is the data about data that you are storing. A set of a name-value pair with which you can store the information regarding an object. Metadata can be assigned to the objects in Amazon S3 bucket.
- **Subresources:** Subresource mechanism is used to store object-specific information.

- **Access control information:** You can put the permissions individually on your files.

Amazon S3 Concepts



- Buckets
- Objects
- Keys
- Regions
- Data Consistency Model

- **Buckets**

- A bucket is a container used for storing the objects.
- Every object is incorporated in a bucket.
- For example, if the object named photos/tree.jpg is stored in the treeimage bucket, then it can be addressed by using the URL <http://treeimage.s3.amazonaws.com/photos/tree.jpg>.
- A bucket has no limit to the amount of objects that it can store. No bucket can exist inside of other buckets.
- S3 performance remains the same regardless of how many buckets have been created.

- The AWS user that creates a bucket owns it, and no other AWS user cannot own it. Therefore, we can say that the ownership of a bucket is not transferrable.
- The AWS account that creates a bucket can delete a bucket, but no other AWS user can delete the bucket.
- **Objects**
 - Objects are the entities which are stored in an S3 bucket.
 - An object consists of object data and metadata where metadata is a set of name-value pair that describes the data.
 - An object consists of some default metadata such as date last modified, and standard HTTP metadata, such as Content type. Custom metadata can also be specified at the time of storing an object.
 - It is uniquely identified within a bucket by key and version ID.
- **Key**
 - A key is a unique identifier for an object.
 - Every object in a bucket is associated with one key.
 - An object can be uniquely identified by using a combination of bucket name, the key, and optionally version ID.
 - For example, in the URL `http://jtp.s3.amazonaws.com/2019-01-31/Amazons3.wsdl` where "jtp" is the bucket name, and key is "2019-01-31/Amazons3.wsdl"
- **Regions**
 - You can choose a geographical region in which you want to store the buckets that you have created.
 - A region is chosen in such a way that it optimizes the latency, minimize costs or address regulatory requirements.
 - Objects will not leave the region unless you explicitly transfer the objects to another region.
- **Data Consistency Model**

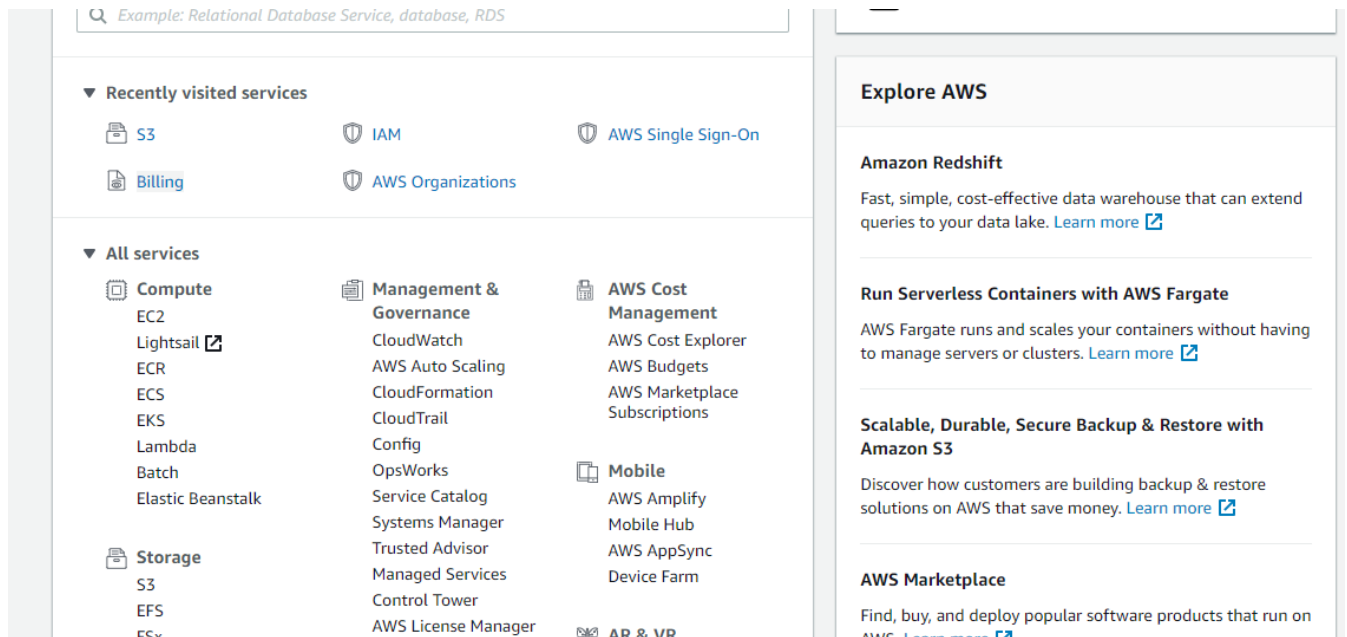
Amazon S3 replicates the data to multiple servers to achieve high availability. Two types of model:

 - **Read-after-write consistency for PUTS of new objects.**

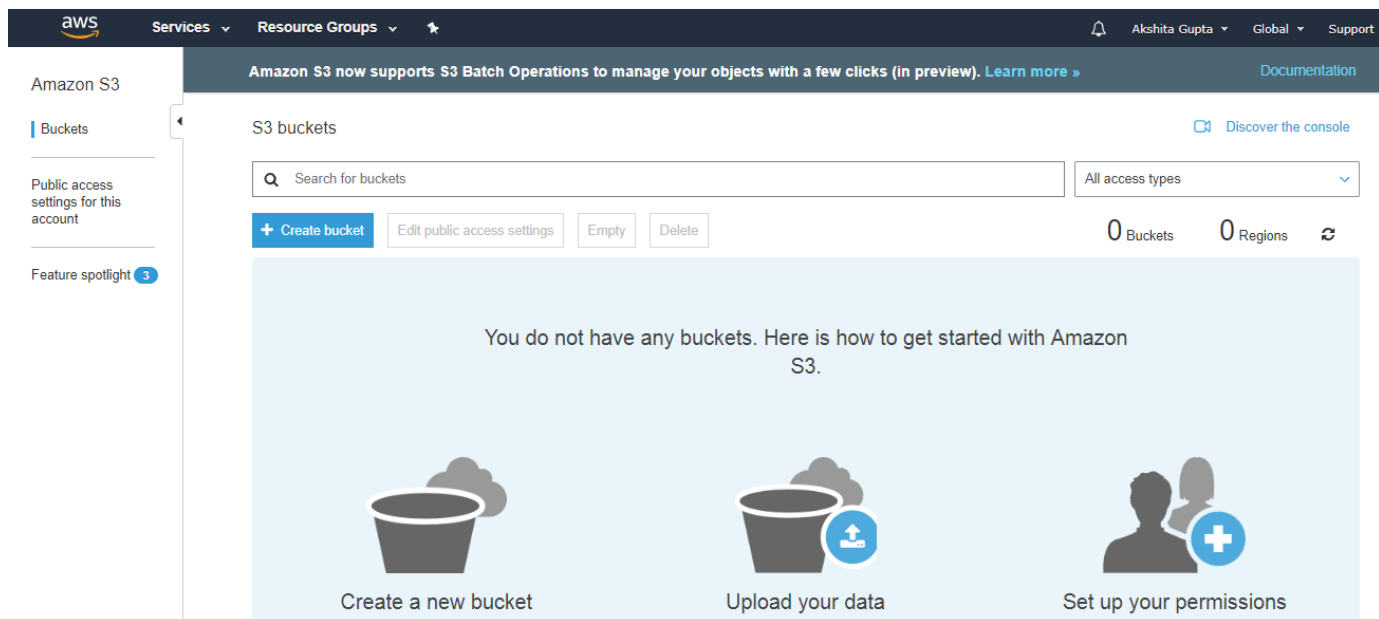
- For a PUT request, S3 stores the data across multiple servers to achieve high availability.
- A process stores an object to S3 and will be immediately available to read the object.
- A process stores a new object to S3, it will immediately list the keys within the bucket.
- It does not take time for propagation, the changes are reflected immediately.
- **Eventual consistency for overwrite PUTS and DELETES**
 - For PUTS and DELETES to objects, the changes are reflected eventually, and they are not available immediately.
 - If the process replaces an existing object with the new object, you try to read it immediately. Until the change is fully propagated, the S3 might return prior data.
 - If the process deletes an existing object, immediately try to read it. Until the change is fully propagated, the S3 might return the deleted data.
 - If the process deletes an existing object, immediately list all the keys within the bucket. Until the change is fully propagated, the S3 might return the list of the deleted key.

Creating an S3 Bucket

- Sign in to the AWS Management console. After sign in, the screen appears is shown below:



- Move to the S3 services. After clicking on S3, the screen appears is shown below:



- To create an S3 bucket, click on the "Create bucket". On clicking the "Create bucket" button, the screen appears is shown below:

aws Services Resource Groups

Akshita Gupta Global Support

Amazon S3

Buckets

Public access settings for this account

Feature spotlight 3

Create bucket

1 Name and region 2 Configure options 3 Set permissions 4 Review

Name and region

Bucket name ⓘ

Enter DNS-compliant bucket name

Bucket name must not be empty

Region

US East (Ohio)

Copy settings from an existing bucket

You have no buckets0 Buckets

Create Cancel Next

- Enter the bucket name which should look like DNS address, and it should be resolvable. A bucket is like a folder that stores the objects. A bucket name should be unique. A bucket name should start with the lowercase letter, must not contain any invalid characters. It should be 3 to 63 characters long.

aws Services Resource Groups

Akshita Gupta Global Support

Amazon S3

Buckets

Public access settings for this account

Feature spotlight 3

Create bucket

1 Name and region 2 Configure options 3 Set permissions 4 Review

Name and region

Bucket name ⓘ

jvatpointbucket

Region

US East (Ohio)

Copy settings from an existing bucket

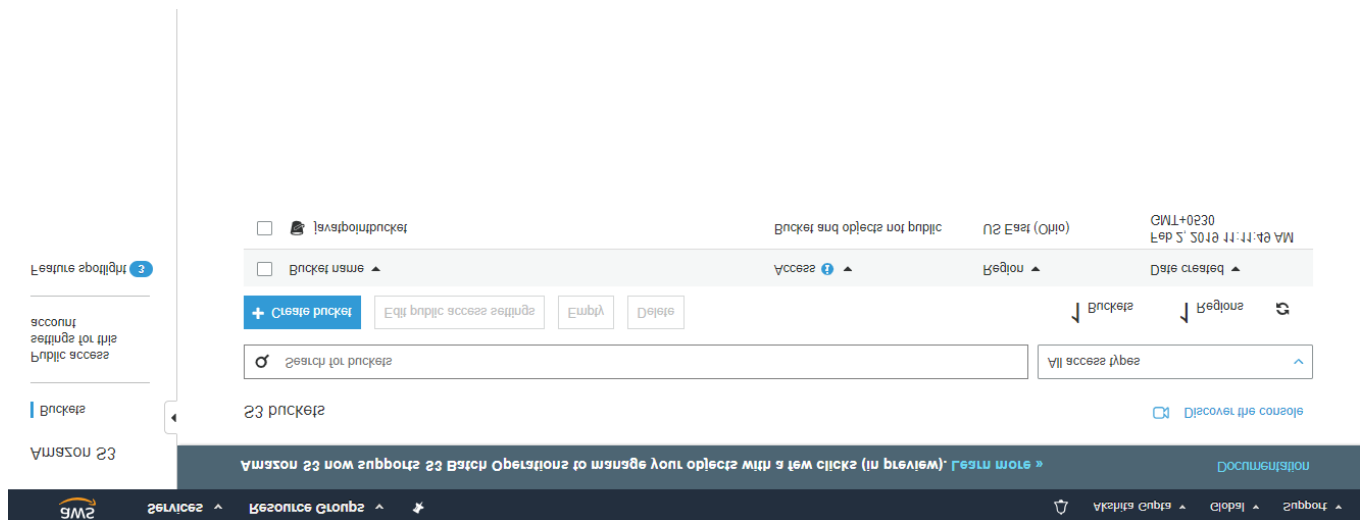
You have no buckets0 Buckets

Create Cancel Next

Feedback English (US)

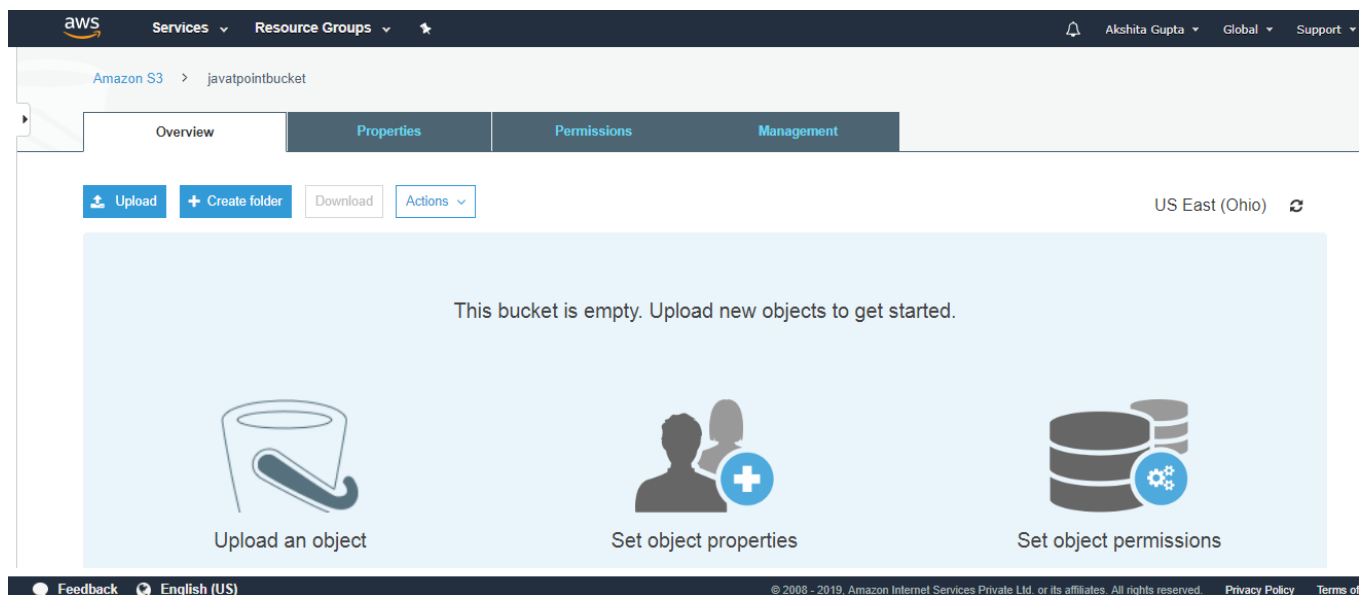
© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms

- Click on the "Create" button. Now, the bucket is created.

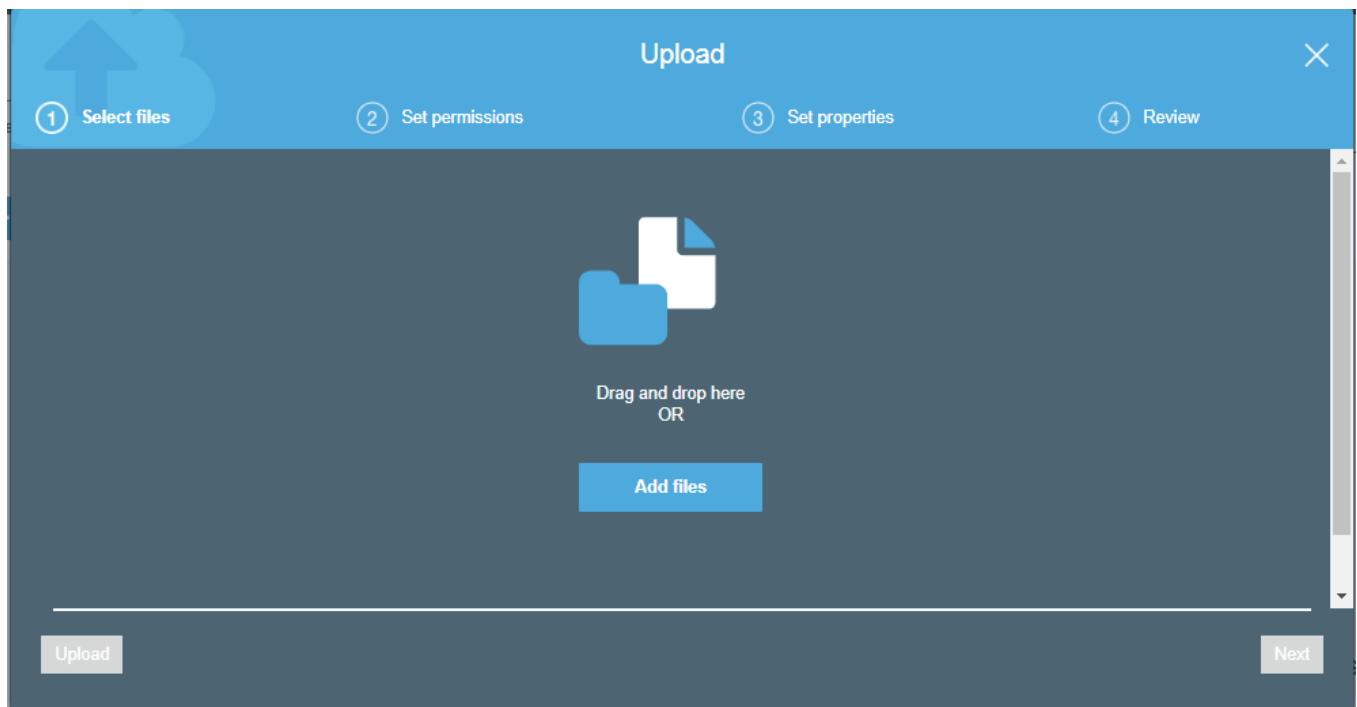


We have seen from the above screen that bucket and its objects are not public as by default, all the objects are private.

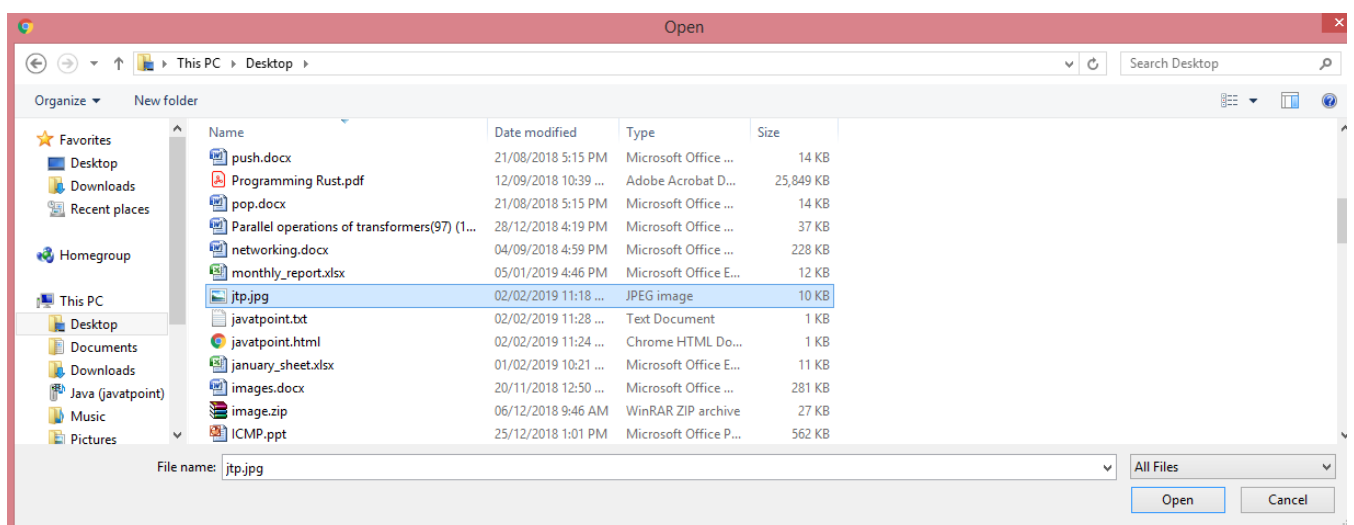
- Now, click on the "javatpointbucket" to upload a file in this bucket. On clicking, the screen appears is shown below:



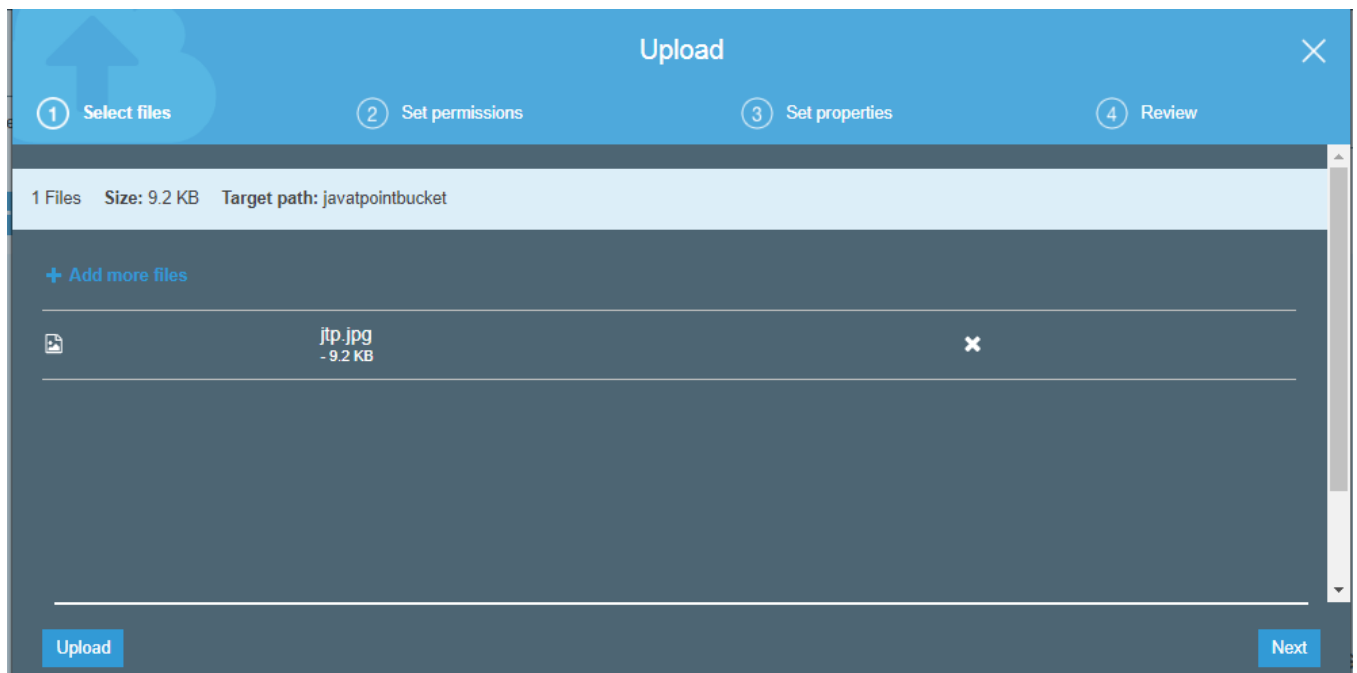
- Click on the "Upload" button to add the files to your bucket.



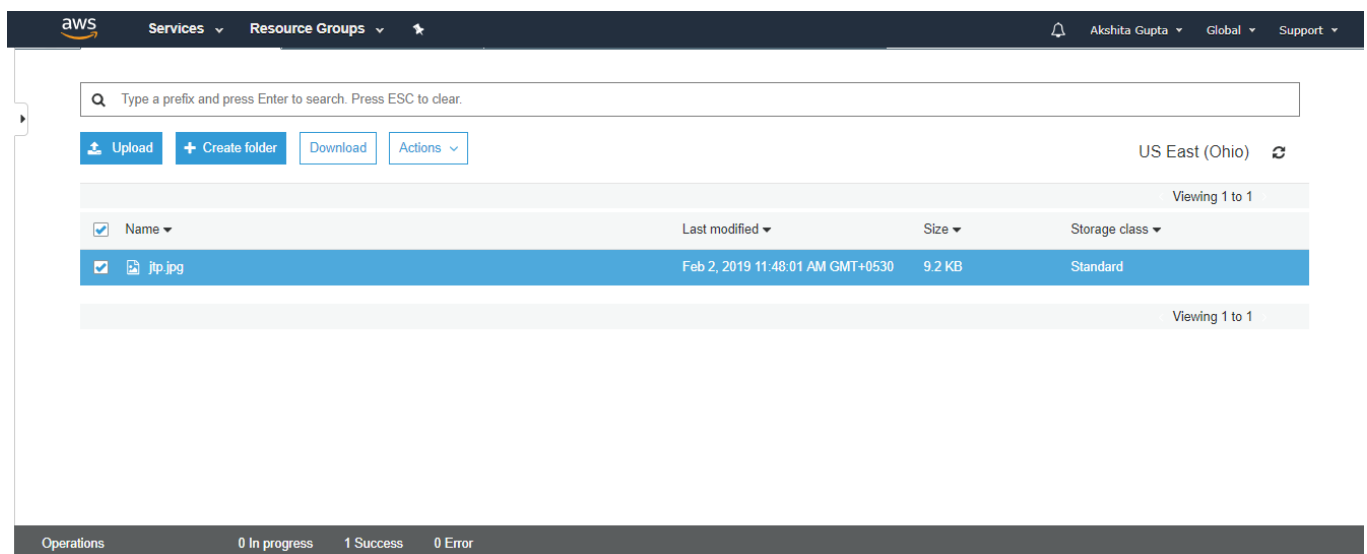
- Click on the "Add files" button.



- Add the **jtp.jpg** file.



- Click on the "upload" button.



From the above screen, we observe that the "jtp.jpg" has been successfully uploaded to the bucket "javatpoint".

- Move to the properties of the object "**jtp.jpg**" and click on the object URL to run the file appearing on the right side of the screen

aws

Services

Resource Groups

Akshita Gupta

Global

Support

Amazon S3

>

javatpointbucket

Overview

Properties

Permissions

Management

Q

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

Upload

Create folder

Download

Actions

Name

Last modified

jtp.jpg

Feb 2, 2019 11:48:01 AM GMT+0530

jtp.jpg

Download

Copy path

Select from

Latest version

Overview

Key

jtp.jpg

Size

9.2 KB

Expiration date

N/A

Expiration rule

N/A

ETag

ddd9f29ed2eaa5229c963f012d4e229f

Last modified

Feb 2, 2019 11:48:01 AM GMT+0530

Object URL

https://s3.us-east-2.amazonaws.com/javatpointbucket/jtp.jpg

Properties

Storage class

Standard

Encryption

None

Metadata

1

Feedback

English (US)

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy

Terms