## Introduction

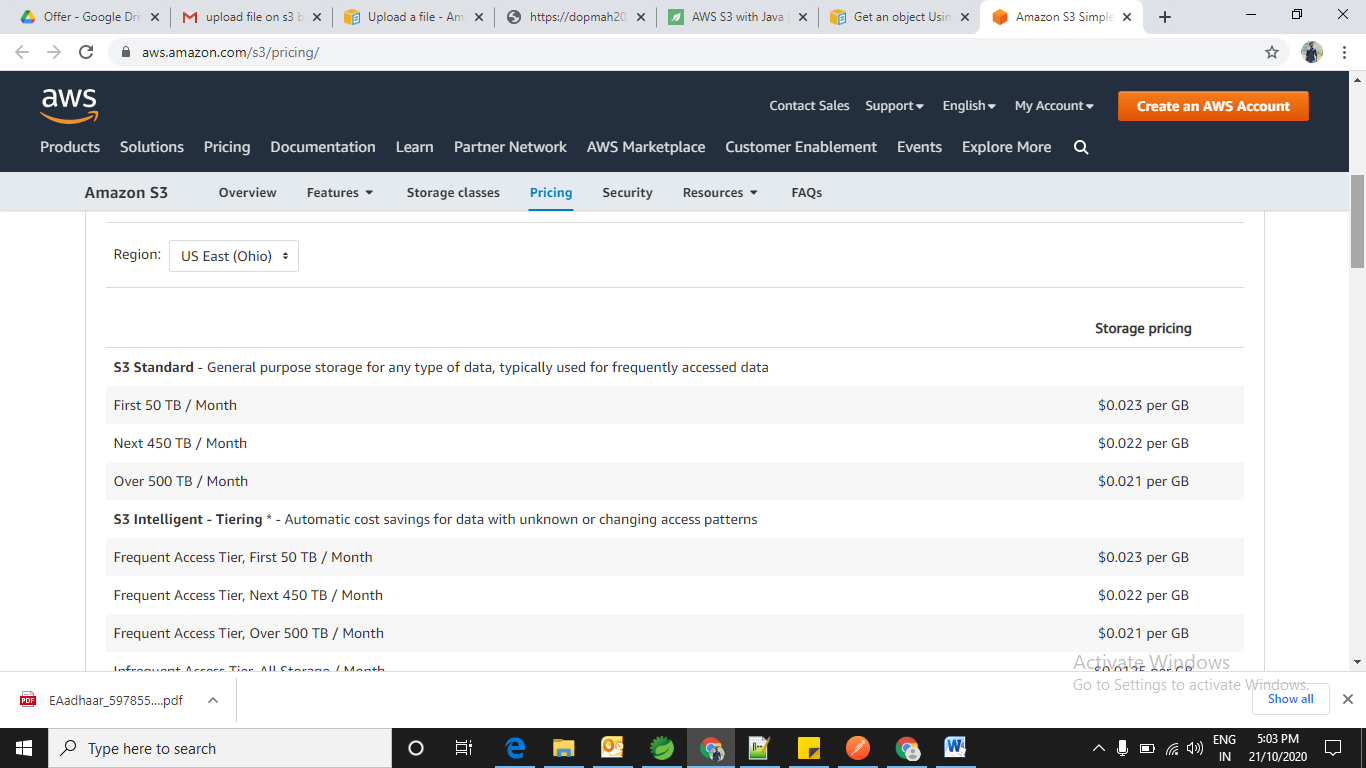
An Amazon **S3(Simple Storage Service) bucket** is a public cloud storage resource available in Amazon Web Services' (**AWS**) it is an object storage offering. Amazon **S3 buckets**, which are similar to file folders, store objects, which consist of data and its descriptive metadata.

## S3 bucket features

* AWS offers several features for Amazon S3 buckets. An IT professional can enable versioning for S3 buckets to preserve every version of an object when an operation is performed on it, such as a copy or delete operation.
* This helps an IT team prevent accidental deletion of an object. Likewise, upon bucket creation, a user can set up server access logs, object-level API logs, tags and [encryption](https://searchsecurity.techtarget.com/definition/encryption).
* Also, S3 Transfer Acceleration helps execute fast, secure transfers from a client to an S3 bucket via AWS [edge locations](https://searchdatacenter.techtarget.com/definition/edge-computing).
* User can specify access **privileges** for the objects stored in a bucket, through mechanisms such as the [AWS Identity and Access Management](https://searchaws.techtarget.com/definition/Amazon-Web-Services-AWS-Identity-and-Access-Management-IAM) service, bucket policies and access control lists.

## S3 bucket pricing

* Pay only for what you use. There is no minimum fee. There are **four cost components** to consider when deciding on which S3 storage class best fits your data profile – storage pricing, request and data retrieval pricing, data transfer and transfer acceleration pricing, and data management features pricing.
* For pricing details visit: <https://aws.amazon.com/s3/pricing/>

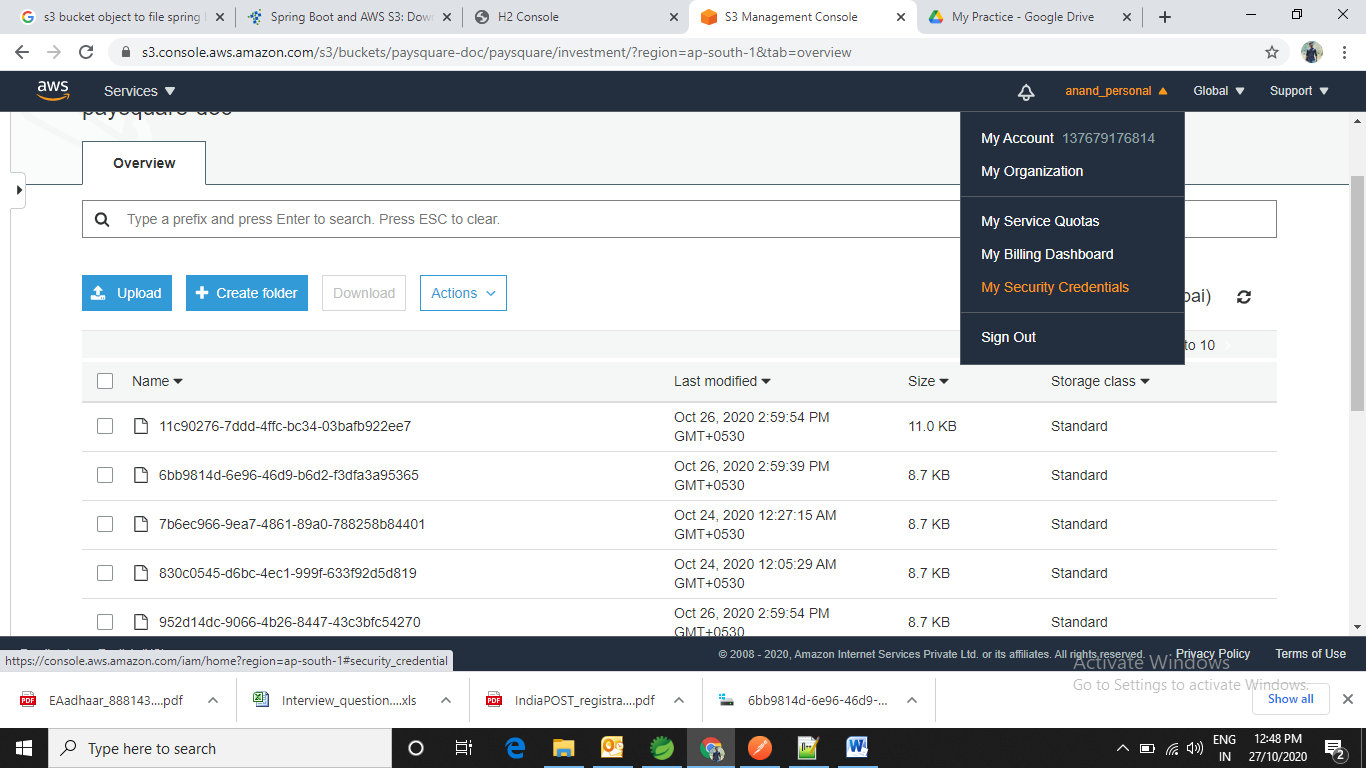


There are three major costs associated with S3:

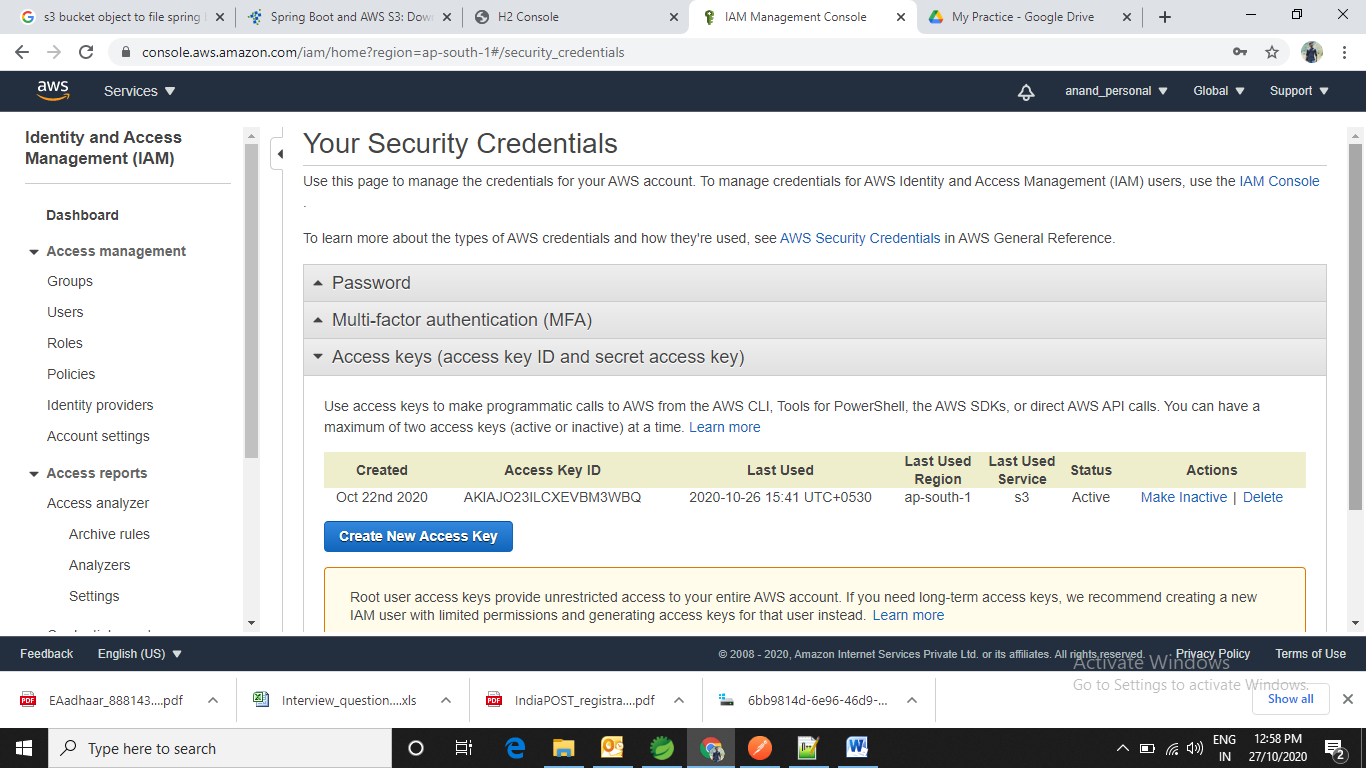
1. **Storage cost**: charged per GB / month. ~ $0.03 / GB / month, charged hourly
2. **API cost for operation of files**: ~$0.005 / 10000 read requests, write requests are 10 times more expensive
3. **Data transfer outside of AWS region**: ~$0.02 / GB to different AWS region, ~$0.06 / GB to the internet.

## Create Credentials For API:

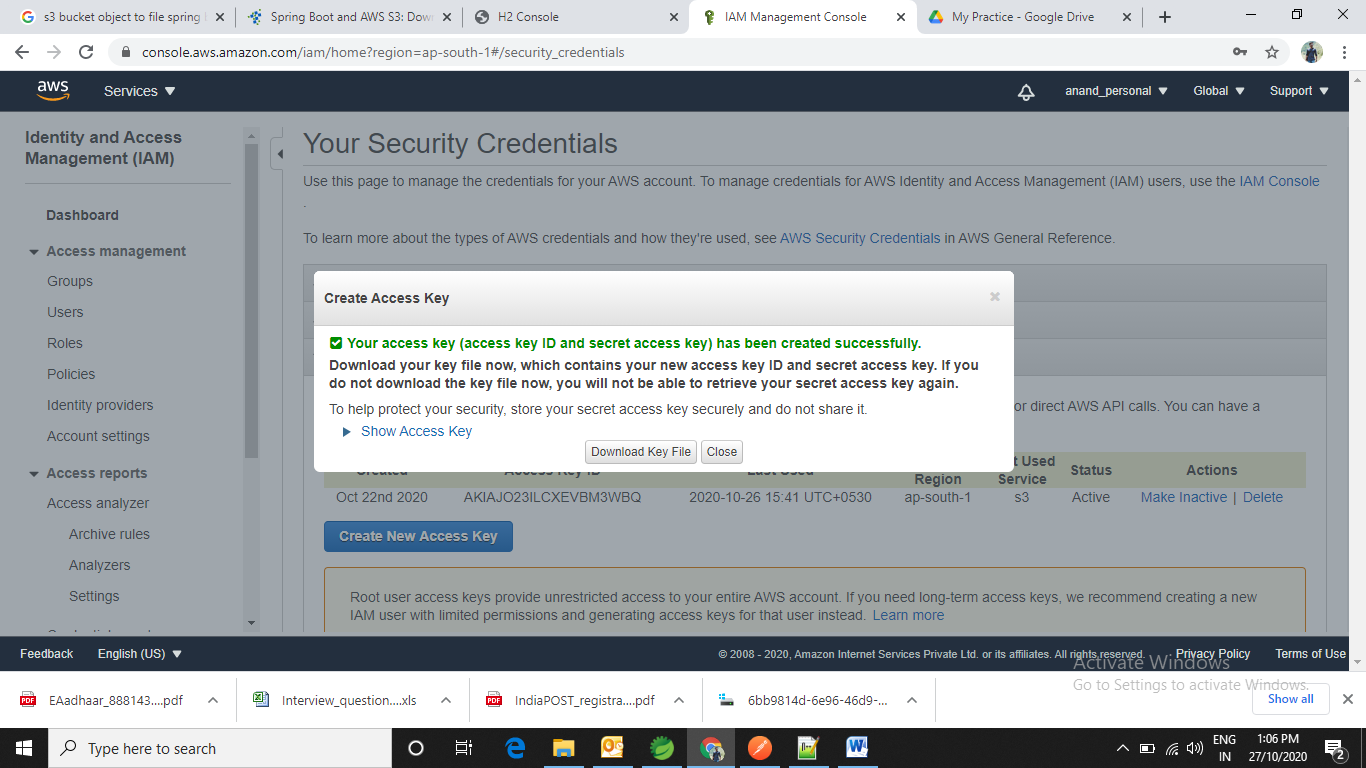
1. Login to AWS account 🡪 Click on your AWS account name 🡪 Click on **My Security Credentials**



1. Click on 🡪 Access keys (access key ID and secret access key) 🡪 Click on **Create New Access Key** button

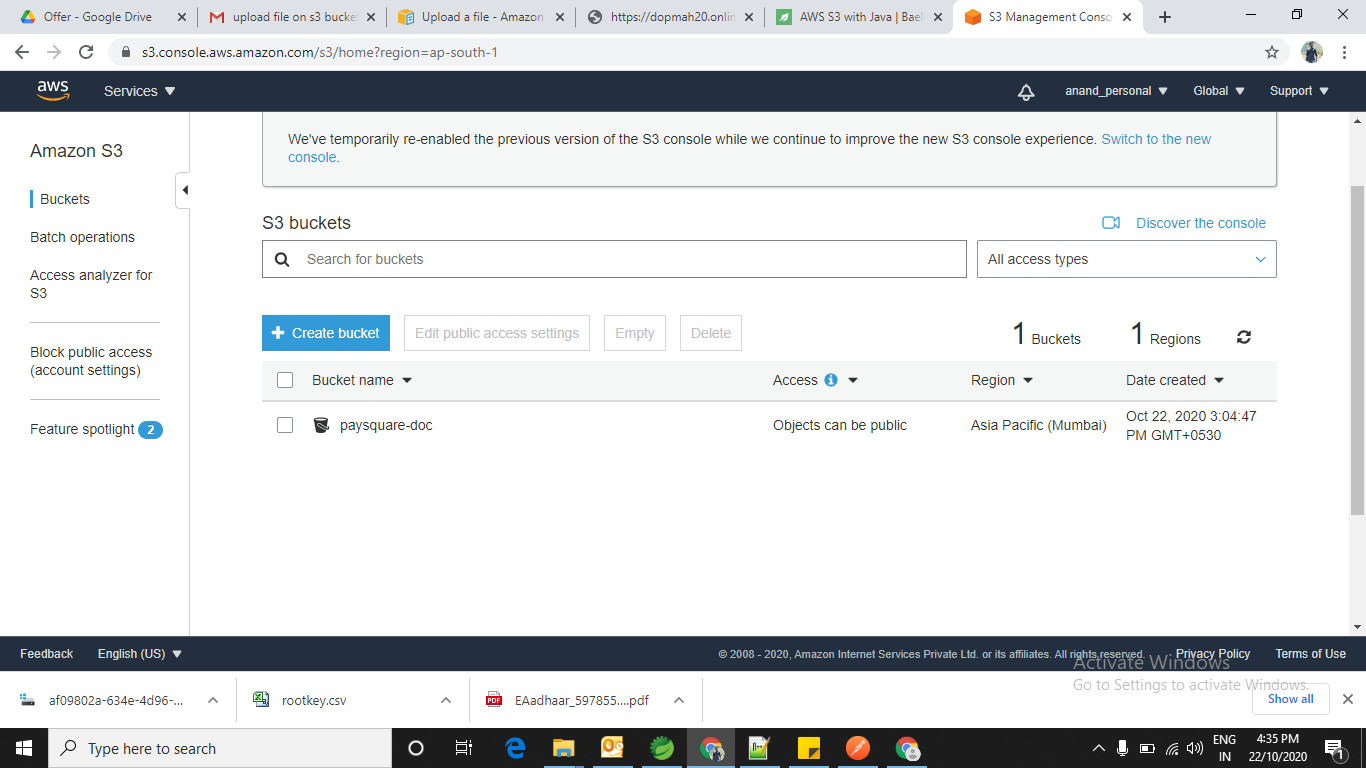


1. After click on Create New Access Key button you will get pop up on screen 🡪 Click on **Show Access Key** from the pop up and copy the information of access key and secret key to your project to access the AWS API.



## Create an S3 Bucket

* Before you can upload data to Amazon S3, you must create a bucket in one of the AWS **Regions** to store your data. After you create a bucket, you can upload an unlimited number of data objects to the bucket.
* The AWS account that creates the bucket by default it owns it.
* You can create up to **100** buckets in each of your **AWS accounts**. If you need additional buckets, you can increase your account bucket quota to a **maximum of 1,000 buckets** by submitting a service quota increase.
* For information about how to increase your bucket quota, see [AWS Service Quotas](https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html)
* To create the buckets visit this [URL](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-bucket.html)*.*
* Log in into AWS account 🡪 Go to services 🡪 S3 🡪 click on Create Bucket



**Upload File (Object) S3 using java**

* Multipart upload allows you to upload a single object as a **set of parts**.
* Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order.
* If transmission of any part fails, you can retransmit that part without affecting other parts.
* After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. In general, when your object size **reaches 100 MB**, you should consider using **multipart uploads** instead of uploading the object in a single operation.
* You need to Saves **the upload ID** that the AmazonS3Client.initiateMultipartUpload() method returns. You provide this upload ID for each subsequent multipart upload operation.
* To upload the file (**object**) up to **5 TB** size using [Multipart Upload API](http://aws.amazon.com/blogs/aws/amazon-s3-multipart-upload/) visit [URL](https://docs.aws.amazon.com/AmazonS3/latest/dev/llJavaUploadFile.html)
* To upload the file (**object**) up to **5 GB** in single request using put request visit [URL](https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-s3-objects.html#upload-object)

**Note:**  In general, when your object size reaches **100 MB**, you should consider using multipart uploads instead of uploading the object in a single operation.

## Access Control List (ACL)-Specific Request Headers

* You can use headers to grant ACL- based permissions. By default, all objects are **private**. Only the owner has full access control.
* When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3.
* These permissions are then added to the ACL on the object. For more information, see [Access Control List (ACL) Overview](https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html) and [Managing ACLs Using the REST API](https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-using-rest-api.html).

## Download the object

* When you download an object through the AWS SDK for Java, Amazon S3 returns all of the object's **metadata** and an input stream from which to read the object's contents.
* To get the uploaded file (object) visit the url [Click Here](https://docs.aws.amazon.com/AmazonS3/latest/dev/RetrievingObjectUsingJava.html)

## Maven Dependency

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk</artifactId>

<version>1.11.163</version>

</dependency>

**References:**

1. <https://www.baeldung.com/aws-s3-java>
2. <https://searchaws.techtarget.com/definition/AWS-bucket#:~:text=An%20Amazon%20S3%20bucket%20is,data%20and%20its%20descriptive%20metadata>.
3. <https://aws.amazon.com/s3/pricing/>
4. <https://calculator.aws/#/createCalculator>
5. <https://docs.aws.amazon.com/AmazonS3/latest/user-guide/using-folders.html>
6. <https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-bucket.html>
7. <https://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html>
8. <https://docs.aws.amazon.com/AmazonS3/latest/dev/llJavaUploadFile.html>
9. <https://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html>
10. <https://docs.aws.amazon.com/AmazonS3/latest/dev/RetrievingObjectUsingJava.html>
11. <https://medium.com/oril/uploading-files-to-aws-s3-bucket-using-spring-boot-483fcb6f8646>
12. <https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html>

**Designed by: Anand Nandeshwar (21-10-2020)**