



# BEL - Amazon Simple Cloud Storage

19.03.2024

---

Shani Kumar Gupta

Backend Engineering Launchpad Cohort - 5

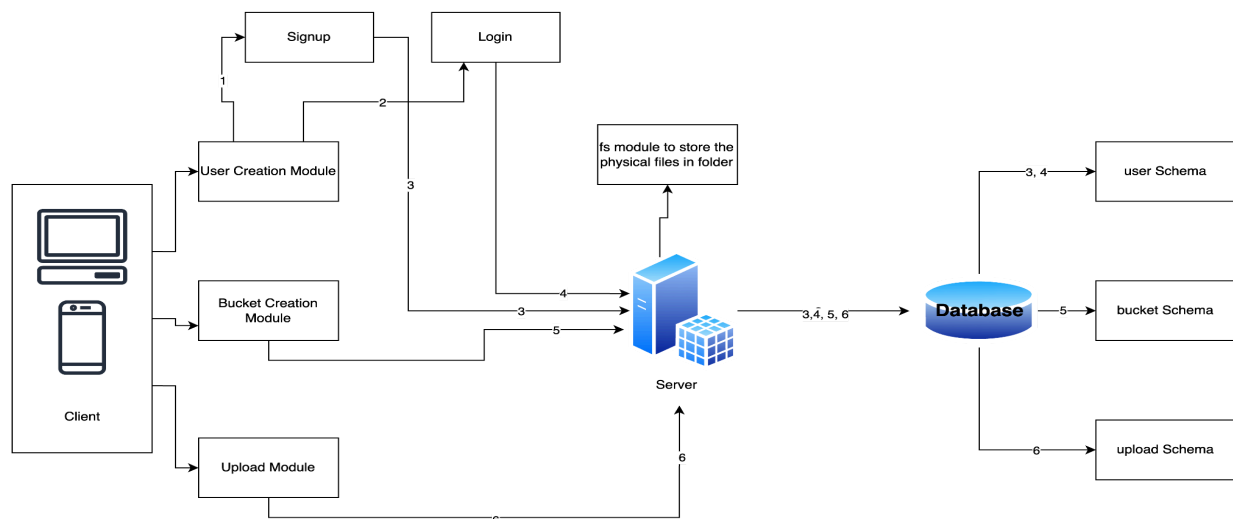
## Overview:

In today's digital age, cloud storage solutions like Amazon S3 are essential for storing, retrieving, and managing vast amounts of data efficiently and securely. For this capstone project, The main goal is to build a simplified clone of Amazon S3, which we'll refer to as "Simple Cloud Storage" (SCS). The SCS will be a web-based storage solution that allows users to store and retrieve files from a personal cloud-based storage space.

## Requirements:

1. User Authentication
2. File Upload and Download
3. File Organization
4. File Listing and Search
5. Permissions and Access Control
6. File Versioning
7. MetaData Management
8. File Duplication (Optional)
9. Usage Analytics (Optional)

## System Design:



## Database Design:

I am going to use the NoSQL database i.e. MongoDB. Please find the initial schema which I am thinking of and could be updated in the future basis on requirements.

### 1) User Schema

```
{
  _id: ObjectId,
  userId: String, (default: 'USER100' + 7 digit unique random alphanumeric
  name: String,
  email: String,
  password: String
}
```

### 2) Bucket Schema

```
{
  _id: ObjectId,
  bucketId: String, (default: 'BUCKET100' + 7 digit unique random alphanumeric
  userId: ObjectId,
  bucketName: String,
  bucketSize: String,
  tags: ["" ]
}
```

### 3) File Schema

```
{
  userId: ObjectId, (ref to userSchema)
  bucketId: ObjectId, (ref to bucketSchema)
  fileName: String,
```

```

    typeOfFile: String,
    filePath: String,
    fileVersion: String,
    tags: [""],
    prevVersionDetails: [{
        fileName: String,
        typeOfFile: String,
        filePath: String,
        fileVersion: String,
        tags: [""],
    }]
}

```

## Key API's:


As per the current approach, I will create the below mentioned API's and will keep on adding the API's if required further.

- 1) simple-cloud-storage/v1/userAuth/login (Method: POST)
- 2) simple-cloud-storage/v1/userAuth/signup (Method: POST)
- 3) simple-cloud-storage/v1/userAuth/list (Method: GET)
- 4) simple-cloud-storage/v1/bucket/createBucket (Method: POST)
- 5) simple-cloud-storage/v1/bucket/fetchBucketList (Method: GET)
- 6) simple-cloud-storage/v1/file/fileUpload (Method: POST)
- 7) simple-cloud-storage/v1/file/fetchUploadFiles (Method: GET) - used to get the list of all files uploaded by the user
- 8) simple-cloud-storage/v1/file/download/:id (Method: POST)
- 9) simple-cloud-storage/v1/file/shareAccess (Method: POST) - Will add all the necessary file details along with the user details whom we want to share the access : (Payload: body)

## Overall Approach:

Technology Used:

1. Frontend (React)

- 
2. Backend (Node and Express)
  3. Database (MongoDB)

In this project, I will use Node and Express to write the server side logic and to store data, will use the MongoDB as database because the file upload data would be huge as well as the upload file is unstructured and can be changed over the time in future.