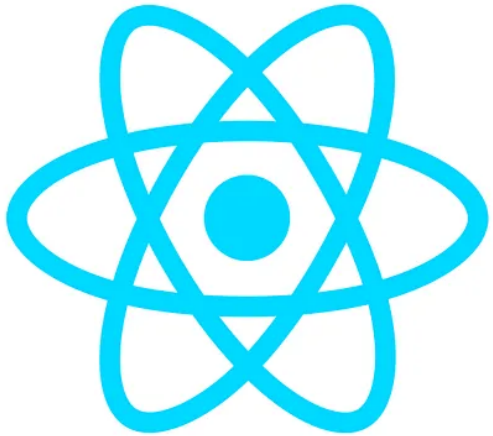
Published in JavaScript in Plain English

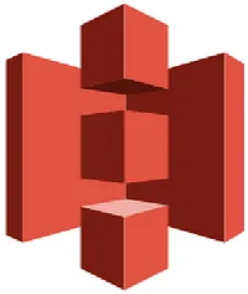
Mohammad Faisal Follow

Feb 17, 2021 · 4 min read · Member-only · Listen

How to Upload Files to AWS S3 in React

In 2 different ways





Credit: Google

Today, with some awesome serverless cloud solutions like AWS and Firebase, uploading a file has become a piece of cake.

Today we will see how we can quickly create an S3 bucket and upload files directly from our front-end React application.

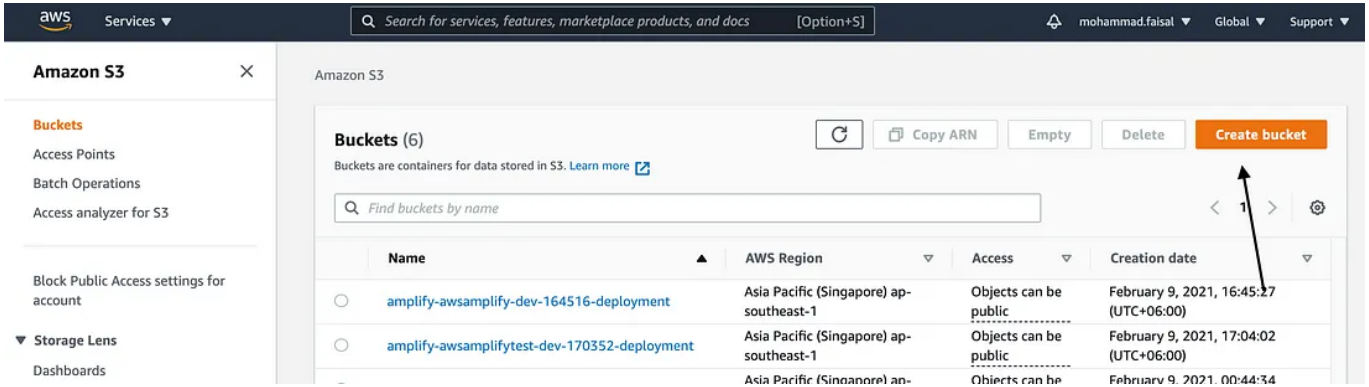
Pre Requisites

1. An AWS Account
2. Basic Understanding of ReactJS

Let's get started!

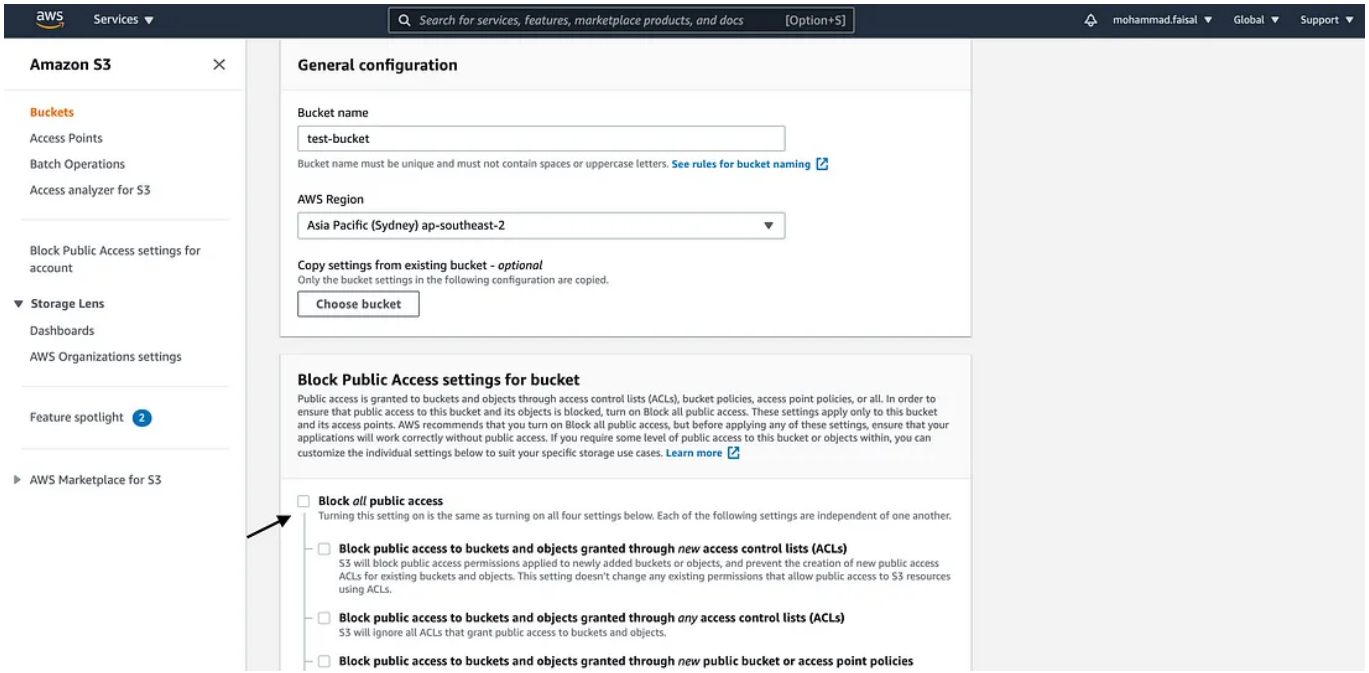
Step 1. Create S3 Bucket

Log in to your aws console. Search for `Amazon S3` and click on `Create bucket`.



aws-console

Then give it a name and select the proper region. Then uncheck the `Block all public access` just for now (You have to keep it unchecked in production).



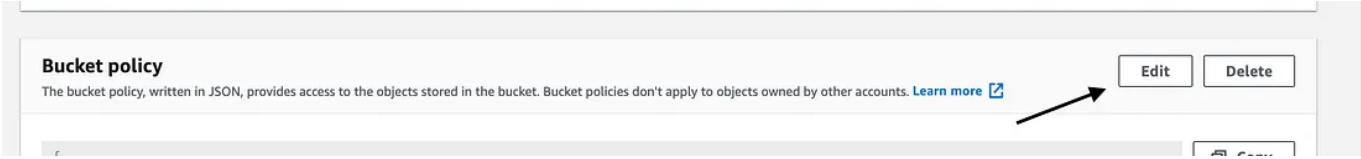
create-s3-bucket

Hit `Create Bucket` and you will see your new bucket on the list.

Step 2. Edit Bucket Policy

Now we will set the bucket policy. Click on the bucket name and go to the `properties` tab.

Scroll down a bit, and you will see a section named `Bucket Policy`



bucket-policy

Click on the `Edit` button. and add the following `json`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicListGet",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:List*",
                "s3:Get*"
            ],
            "Resource": [
                "arn:aws:s3:::BUCKET_NAME",
                "arn:aws:s3:::BUCKET_NAME/*"
            ]
        }
    ]
}
```

s3-bucket-policy hosted with ❤ by GitHub view raw

This will allow public access to the folder's content for now.

Step 3. Edit CORS Policy

Scroll down a bit further, and you will get a section where you can edit the `CORS` policy. As we will be uploading files from another endpoint, we need to ensure that our bucket does not block that for `cors`

cors-policy

Then add the following `.json` file to the policy

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE",
            "GET"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

cors-policy hosted with ❤️ by GitHub view raw

Now we are ready to upload files to our AWS S3 bucket. Now moving on with the frontend part.

Step 4. Set Up React Project

To follow along, you will need a basic React project setup. You can do it easily by

```
npx create-react-app s3-upload
```

It will scaffold a basic application for you.

Step 5. Upload File Via Native SDK

If we use other aws services in our project, then we have a special npm package for that named `aws-sdk` . We can get it by

```
yarn add aws-sdk
```

Then create a new component named `UploadImageToS3WithNativeSdk` and add the following code there

```
1  import React ,{useState} from 'react';
2  import AWS from 'aws-sdk'
3
4  const S3_BUCKET ='YOUR_BUCKET_NAME_HERE';
5  const REGION ='YOUR_DESIRED_REGION_HERE';
6
```

```
AWS.config.update({
    accessKeyId: 'YOUR_ACCESS_KEY_HERE',
    secretAccessKey: 'YOUR_SECRET_ACCESS_KEY_HERE'
})

const myBucket = new AWS.S3({
    params: { Bucket: S3_BUCKET},
    region: REGION,
})

const UploadImageToS3WithNativeSdk = () => {

    const [progress , setProgress] = useState(0);
    const [selectedFile, setSelectedFile] = useState(null);

    const handleFileInput = (e) => {
        setSelectedFile(e.target.files[0]);
    }

    const uploadFile = (file) => {

        const params = {
            ACL: 'public-read',
            Body: file,
            Bucket: S3_BUCKET,
            Key: file.name
        };

        myBucket.putObject(params)
            .on('httpUploadProgress', (evt) => {
                setProgress(Math.round((evt.loaded / evt.total) * 100))
            })
            .send((err) => {
```

```
import React , {useState} from 'react';
import { uploadFile } from 'react-s3';


const S3_BUCKET ='YOUR_BUCKET_NAME';
const REGION ='YOUR_REGION_NAME';
const ACCESS_KEY ='YOUR_ACCESS_KEY';
const SECRET_ACCESS_KEY ='YOUR_SECRET_ACCESS_KEY';

const config = {
    bucketName: S3_BUCKET,
    region: REGION,
    accessKeyId: ACCESS_KEY,
    secretAccessKey: SECRET_ACCESS_KEY,
}

const UploadImageToS3WithReactS3 = () => {

    const [selectedFile, setSelectedFile] = useState(null);

    const handleFileInput = (e) => {
        setSelectedFile(e.target.files[0]);
    }

    const handleUpload = async (file) => {
        uploadFile(file, config)
            .then(data => console.log(data))
            .catch(err => console.error(err))
    }

    return <div>
        <div>React S3 File Upload</div>
        <input type="file" onChange={handleFileInput}/>
        <button onClick={() => handleUpload(selectedFile)}> Upload to S3</button>
```

```
35      </div>
36  }
37
38  export default UploadImageToS3WithReactS3;
```

UploadImageToS3WithReactS3.ts hosted with ❤️ by GitHub view raw

And also, don't forget to change the parameter values.

Conclusion

And there you go! Here is how you can upload any file to an s3 bucket.

Explore the documentation for more customization according to your need.

UPDATE:

Read the below article to learn how to do it securely.

How to Access Private S3 Buckets Securely

Using pre-signed URL

javascript.plainenglish.io

Have a great day! :D

Get in touch with me via LinkedIn or my Personal Website.

21 Best Practices for a Clean React Project

Practical advice for improving code quality

betterprogramming.pub

How To Use AWS DynamoDB in React

Let's try to access DynamoDB from a React application

betterprogramming.pub

20 Essential Parts Of Any Large-Scale React App

If you're writing enterprise-level code, you need to know this

javascript.plainenglish.io

45 NPM Packages to Solve 16 React Problems

An in-depth guide on how to choose the perfect npm package

javascript.plainenglish.io

Programming JavaScript AWS React Software Development

Enjoy the read? Reward the writer.[Beta]

Your tip will go to Mohammad Faisal through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Get my articles directly into your inbox

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

Subscribe

About Help Terms Privacy