**Deployment Notes for react app to aws s3**

process of creating a simple React application and deploying it to Amazon Web Services (AWS) S3, a popular and cost-effective way to host static websites

Prerequisites: Before we get started, make sure you have the following prerequisites:

1. [Node.js](https://nodejs.org/) and npm (Node Package Manager) installed on your computer.
2. An AWS account. If you don’t have one, you can [sign up here](https://aws.amazon.com/).

**Step 1: Create a React App**

Let’s start by creating a simple React application. Open your terminal and run the following commands:

npx create-react-app simple-react-app

cd simple-react-app

npm start

This will create a new React app, set up the development environment, and start the development server. You can access your app at [http://localhost:3000](http://localhost:3000/).

**Step 2: Build the React App**

To prepare your app for deployment, you need to build it. In your terminal, run:

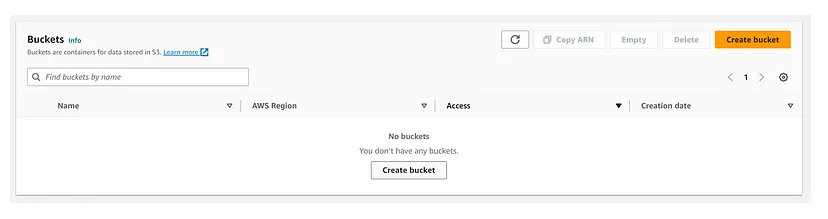
npm run build

This will generate a build folder in your project directory with optimized production-ready files.

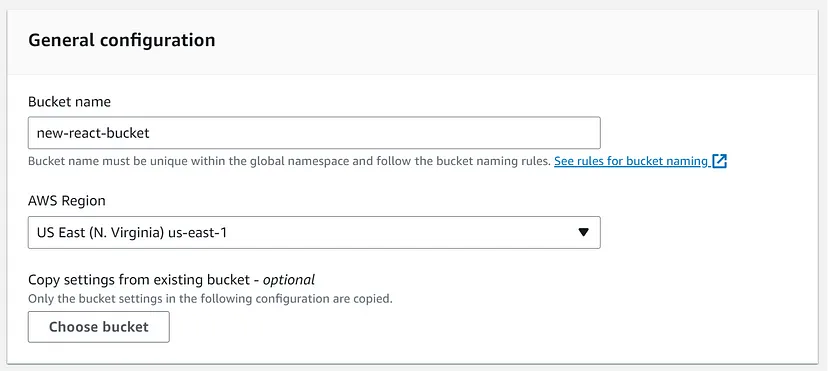
**Step 3: Create an AWS S3 Bucket**

Now, let’s create an S3 bucket to host our React app. Follow these steps:

* Log in to your AWS Management Console.
* Navigate to the S3 service.
* Click the “Create bucket” button.



* Choose a unique name for your bucket, following AWS naming conventions. Make a note of this name; you’ll need it later.
* Select a region for your bucket. Typically, you should choose a region geographically close to your target audience for faster access.

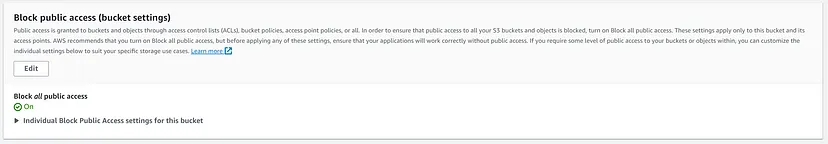


* Leave all other settings as their defaults and create the bucket.

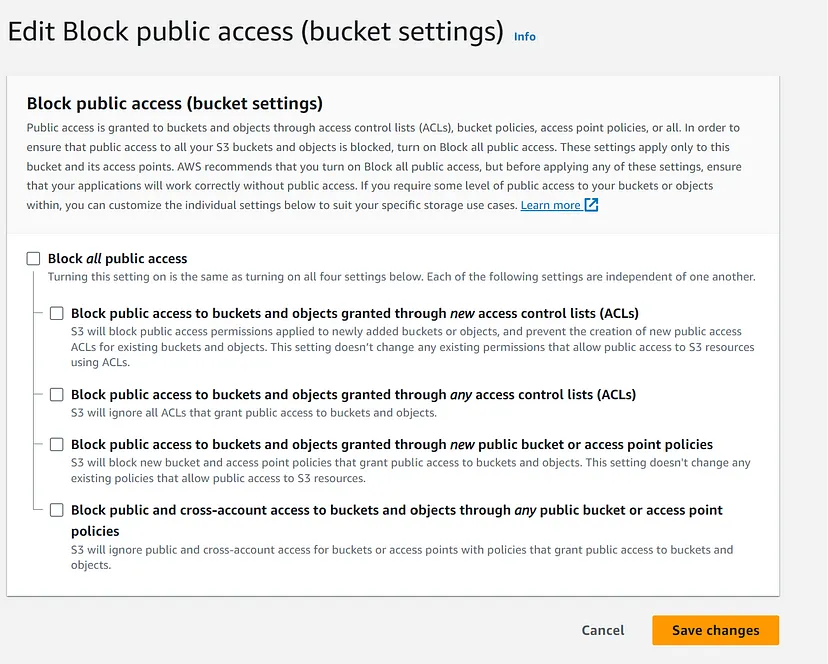
**Step 4: Configure Bucket Permissions**

For your React app to be publicly accessible, you’ll need to configure the bucket’s permissions. Follow these steps:

* Select your newly created bucket in the S3 dashboard.
* Click on the “Permissions” tab.
* Under “Block public access,” click “Edit.”



* Uncheck all options for public access and save your changes.



* Go to the “Bucket Policy” tab and add a bucket policy that grants public read access to your bucket. Replace <YOUR\_BUCKET\_NAME> with your actual bucket name.
* Save your changes.

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::<YOUR\_BUCKET\_NAME>/\*"

}

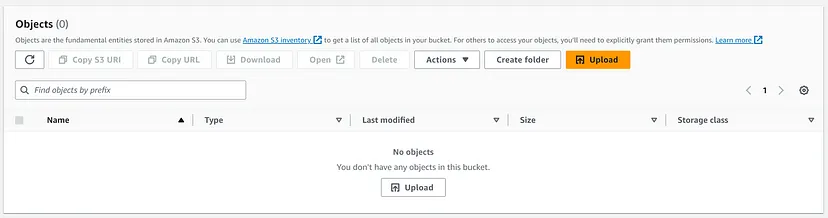
]

}

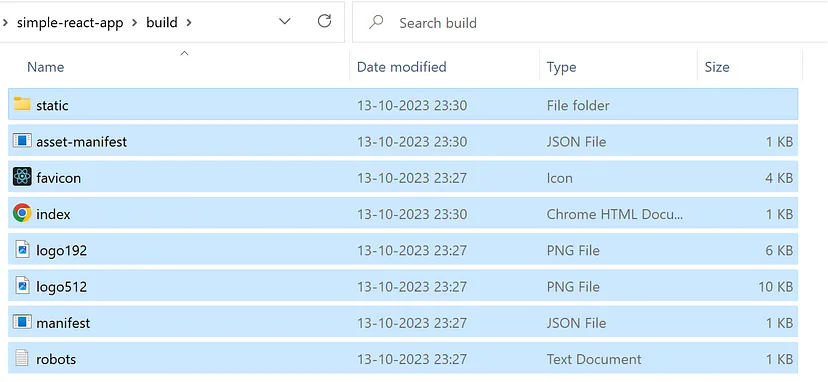
**Step 5: Upload Your React App to S3**

You can use the AWS CLI or the AWS S3 web interface to upload your React app to the S3 bucket. Here’s how to do it using the web interface:

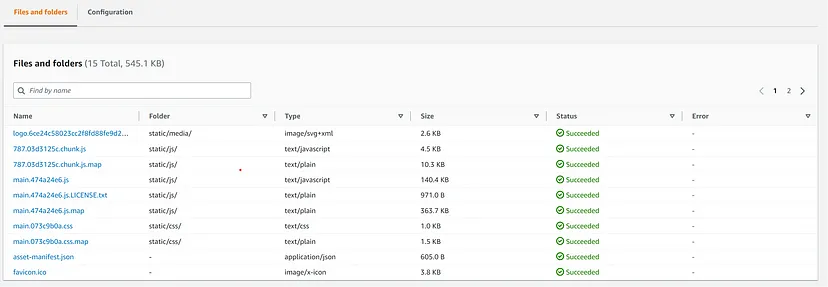
* Go to the “Objects” tab of your S3 bucket.
* Click the “Upload” button.



* Select all the files and folders from your build folder.



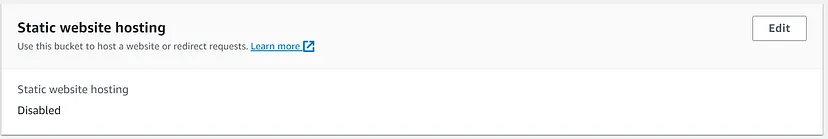
* Click “Next,” and then “Upload” to add your app to the bucket.



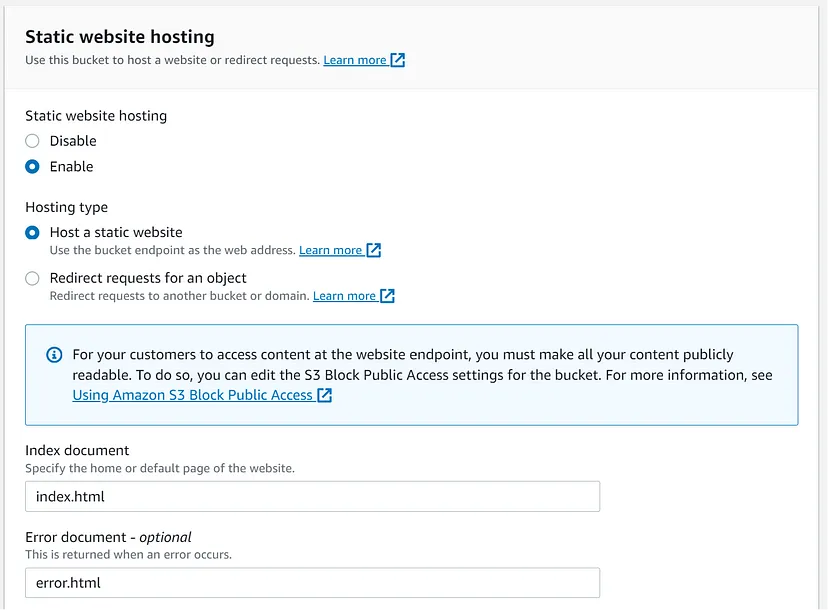
**Step 6: Configure Static Website Hosting**

To make your S3 bucket host a static website, follow these steps:

* In the S3 dashboard, go to the “Properties” tab of your bucket.
* Scroll down to the “Static website hosting” card and click “Edit.”



* Choose “Host a static website.”
* Enter “index.html” for the Index document and “error.html” for the Error document. These are the default filenames that React expects.
* Save your changes.



**Step 7: Access Your Deployed React App**

After configuring static website hosting, you’ll see an “Endpoint” URL in the “Static website hosting” card. It should look like this: http://<YOUR\_BUCKET\_NAME>.s3-website-<AWS\_REGION>.amazonaws.com.

