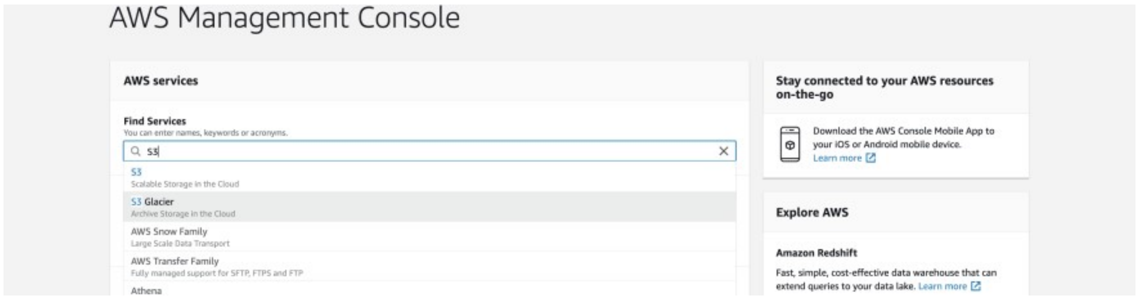
Create My React App in s3 AWS-

Create Amazon S3 Bucket

First, log in to your AWS account. On the AWS Management Console, click S3 from the list of services under the Storage section or use the search bar.



AWS Management Console

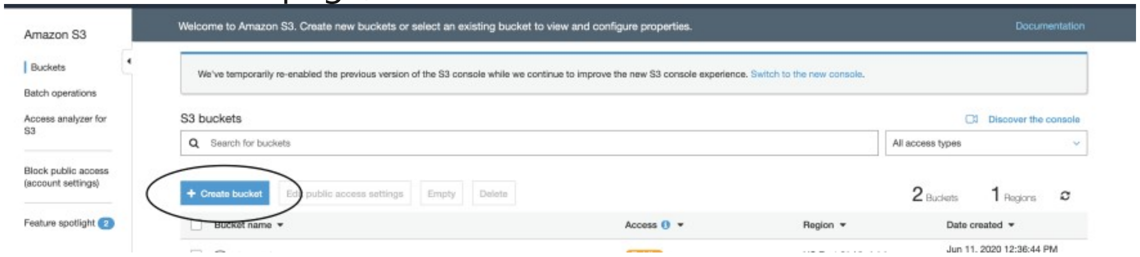
AWS services

Find Services
You can enter names, keywords or acronyms.

s3

S3
Scalable Storage in the Cloud
S3 Glacier
Archive Storage in the Cloud
AWS Snow Family
Large Scale Data Transport
AWS Transfer Family
Fully managed support for SFTP, FTPS and FTP
Athena
Stay connected to your AWS resources on-the-go
Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more
Explore AWS
Amazon Redshift
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

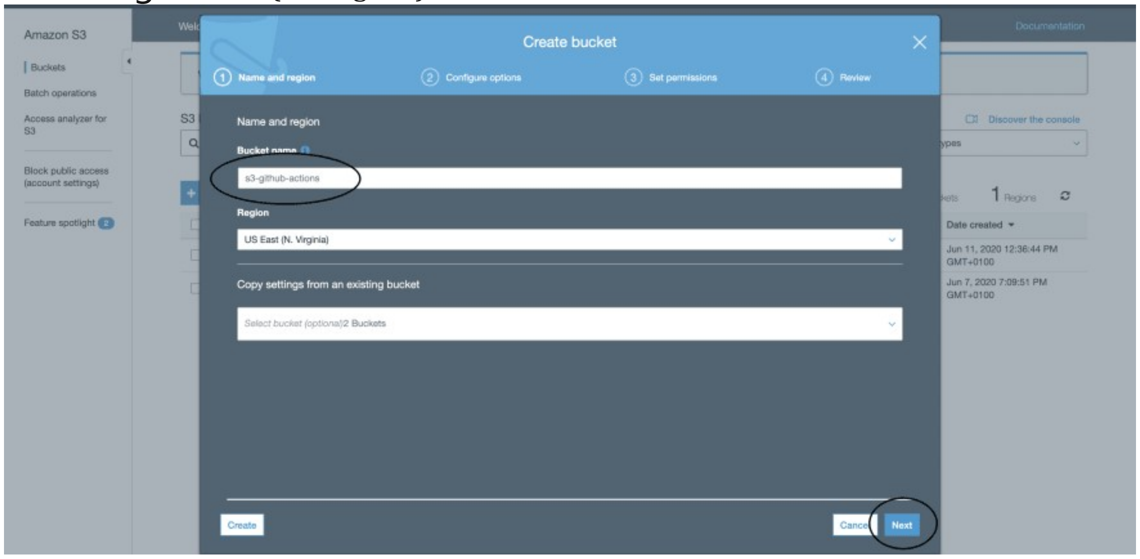
On the Amazon S3 page, click on Create Bucket



Welcome to Amazon S3. Create new buckets or select an existing bucket to view and configure properties.

We've temporarily re-enabled the previous version of the S3 console while we continue to improve the new S3 console experience. Switch to the new console.

To create a bucket, provide a Bucket Name. An S3 bucket name must be unique amid all buckets universally in Amazon S3. Also, take note of the Region you are creating the bucket in. For this post, we are using US East (N. Virginia) which is us-east-1.



Create bucket

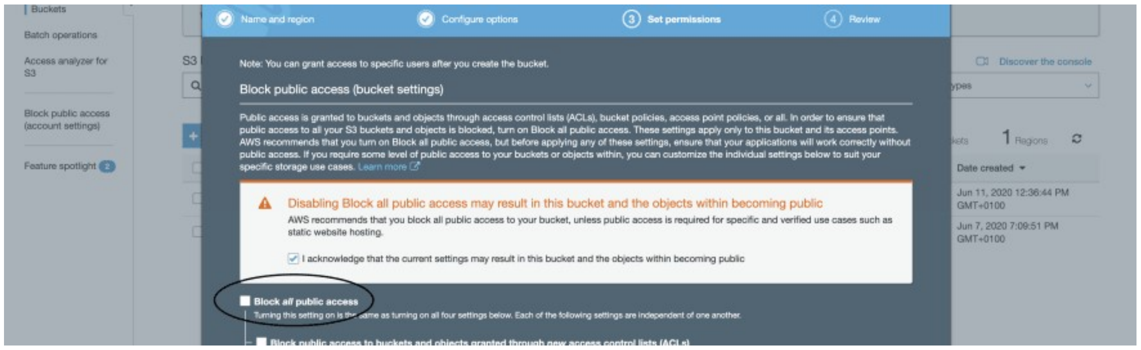
Name and region

Bucket name: s3-github-actions

Region: US East (N. Virginia)

Copy settings from an existing bucket: Select bucket (optional)2 Buckets

Uncheck the checkbox for Block all public access. After, click on Next and Review bucket configurations. Then click Create bucket.



Note: You can grant access to specific users after you create the bucket.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Disabling Block all public access may result in this bucket and the objects within becoming public

AWS recommends that you block all public access to your bucket, unless public access is required for specific and verified use cases such as static website hosting.

✓ I acknowledge that the current settings may result in this bucket and the objects within becoming public

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through new access control lists (ACLs)

Jun 11, 2020 12:36:44 PM GMT+0100

Jun 7, 2020 7:09:51 PM GMT+0100

Add Bucket Policy

This makes the contents of your bucket publicly available. This action is not recommended when working with S3 buckets, but for this our purpose this is fine.

Under Buckets, choose the name of your bucket(s3-github-actions) > Choose Permissions > Choose Bucket Policy.

Copy the following bucket policy, and paste it in the editor.

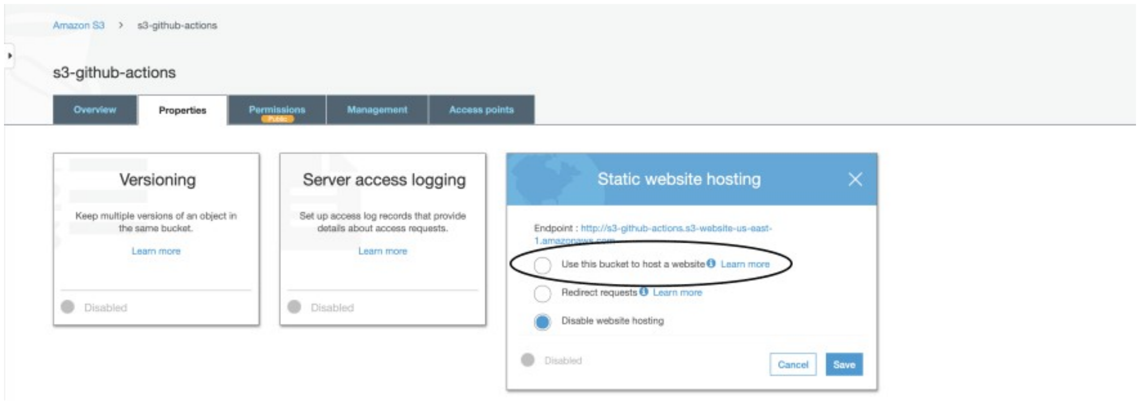
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Update the snippet to include your bucket name. In the bucket policy, <bucket-name> you must update this name to match your

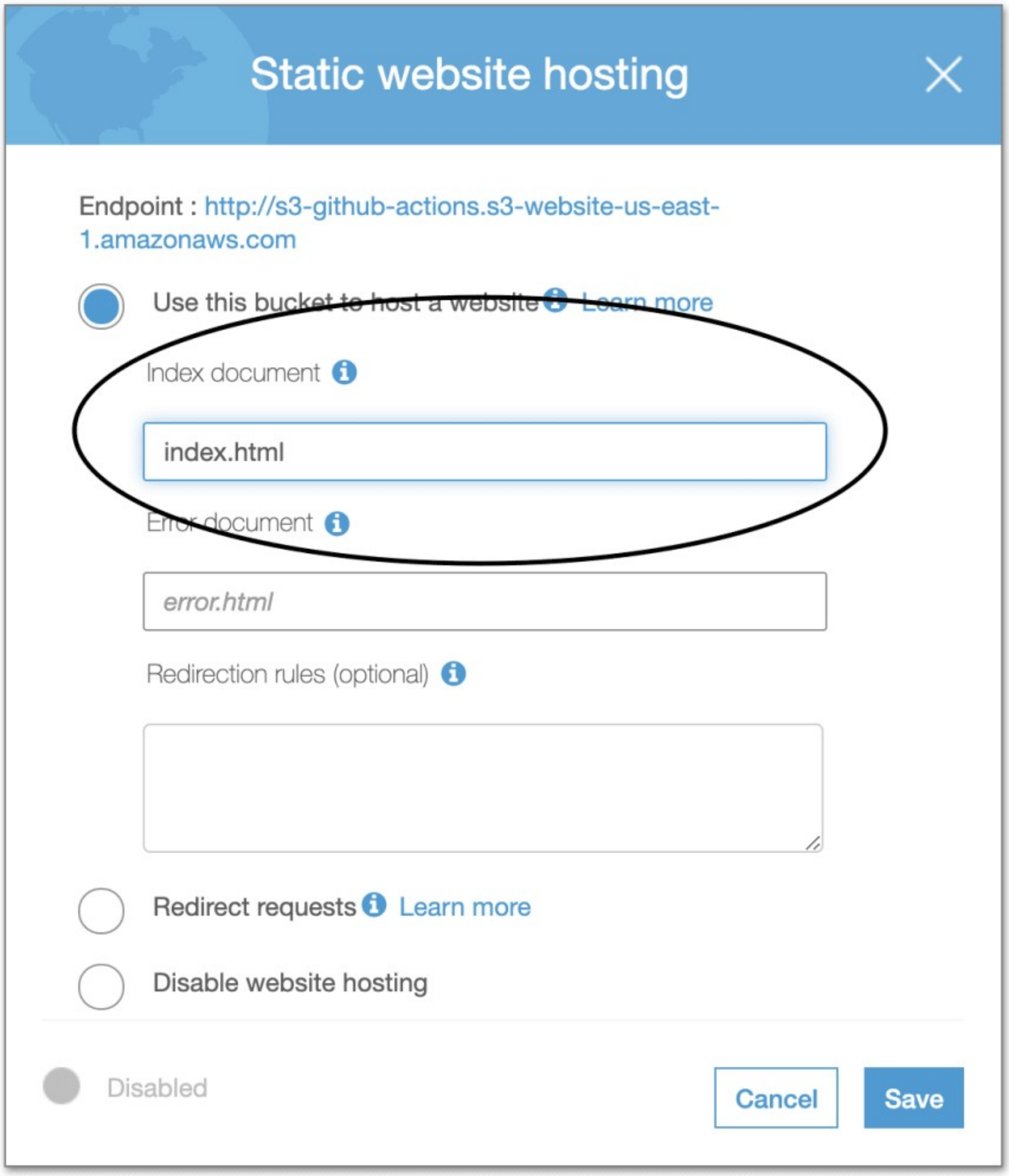
bucket name.
Then, click on Save.

Enable Static Website Hosting

Click Use this bucket to host a website.


Amazon S3 > s3-github-actions
s3-github-actions
Overview
Properties
Permissions
Management
Access points
Versioning
Keep multiple versions of an object in the same bucket.
Learn more
Disabled
Server access logging
Set up access log records that provide details about access requests.
Learn more
Disabled
Static website hosting
Endpoint : http://s3-github-actions.s3-website-us-east-1.amazonaws.com
Use this bucket to host a website Learn more
Redirect requests Learn more
Disable website hosting
Disabled
Cancel
Save

Type index.html in the Index document field and Save.



Endpoint : http://s3-github-actions.s3-website-us-east-1.amazonaws.com

Use this bucket to host a website

Index document: index.html

Error document: error.html

Redirection rules (optional)

Redirect requests

Disable website hosting

Note: Take note of the Endpoint URL, our website will be accessible in the browser using this URL.

Create and Push React App to GitHub

Now we have our S3 bucket, it's time to create and push our React App to GitHub.

- First, create a New Repository on GitHub.

After creating a repository, You could:

- Create a React application using Create React App or Parcel-Bundler and ensure that there is a build script in the package.json file will output to a dist folder.

```
$ git init # initialize git locally
$ git add . # add changes to git
$ git commit -m "React App" # commit changes
$ git remote add origin <your-github-repo-url.git> # add remote origin
$ git push -u origin master # push to remote master branch
```

OR

- Clone the sample React App repository S3-Github Actions React App that we will be using for this post and add your repository's remote URL. GitHub Repo - S3-Github Actions React App

To add new remote (this will add a new remote called actions):

```
$ git remote add actions <your-github-repo-url.git> # add remote actions
$ git push -u actions master # push to remote master branch
```

To set up our workflow, we need to provide the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY and AWS_REGION of the S3 bucket in other to connect successfully to Amazon S3.

Get AWS Authorization

On the AWS Console:

1. Click on IAM under the Security, Identity, & Compliance section.
2. Click on Users and select your preferred user.
3. Under Security Credentials, click on Create Access Key. This will create an AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY, copy these values. You can also manage key access by either deleting or making it inactive.

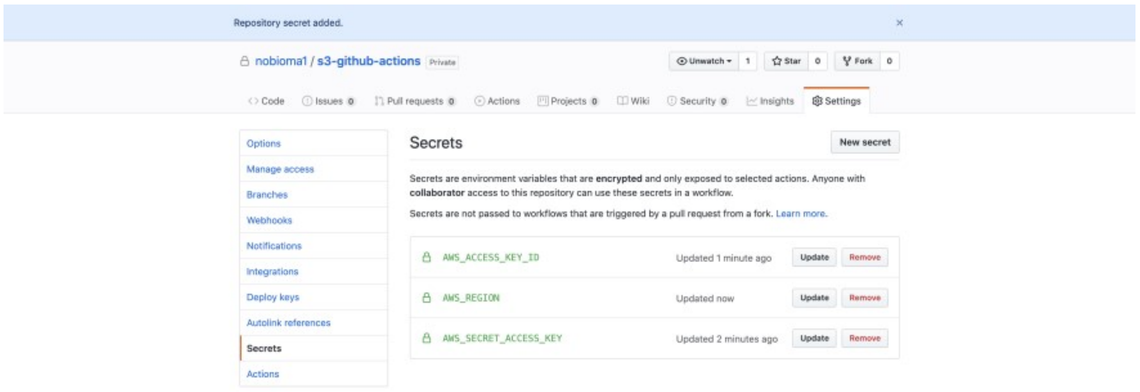
Even with a confidante, you do not ever what to share your access keys. Your confidante might have a confidante. Who knows! 🤷.

So that's why we will be passing some very important values as Secrets on GitHub then later access them in the workflow file using the expression syntax. ${{ <expression> }}

Back to Github

Click on the Settings tab, Select Secret on the left menu, then click on New Secret to add a secret providing the Name and Value.

Name	Value
AWS_ACCESS_KEY_ID	your-aws-access-key-id
AWS_SECRET_ACCESS_KEY	your-aws-secret-access-key
AWS_REGION	us-east-1 or your-aws-s3-region



Secrets

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

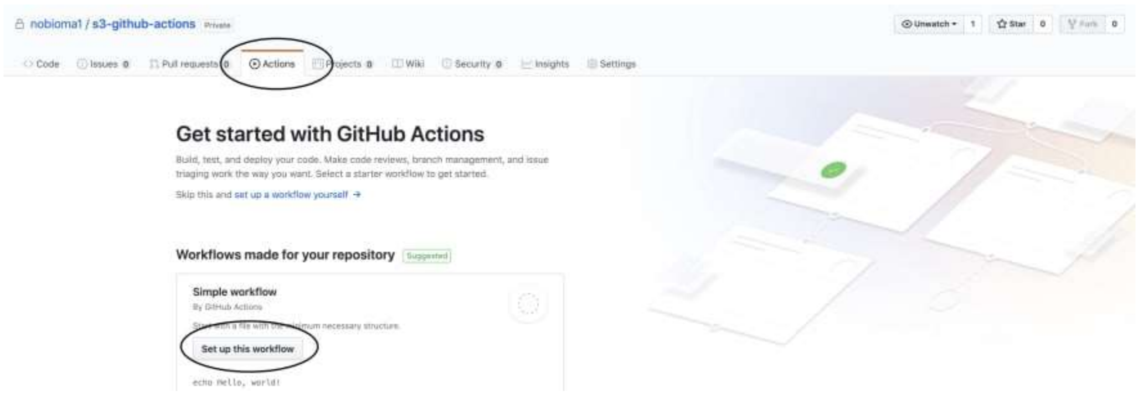
Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

AWS_ACCESS_KEY_ID	Updated 1 minute ago
AWS_REGION	Updated now
AWS_SECRET_ACCESS_KEY	Updated 2 minutes ago

Setup Github Actions

Now, we have the S3 bucket set up and a React app to deploy.

On the GitHub repository, click on the Actions tab to open the Github actions page. On the Actions page, click on the Set up this workflow or set up a workflow yourself -> button, this will redirect to a new page with a web editor containing some boilerplate code but we will get rid of that.


nobioma1 / s3-github-actions Private
Unwatch 1 Star 0 Fork 0
Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a starter workflow to get started.

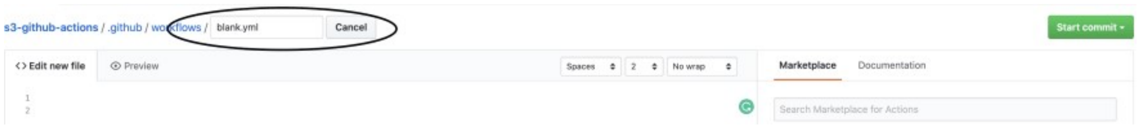
Skip this and set up a workflow yourself →

Workflows made for your repository Suggested

Simple workflow
By GitHub Actions
Start with a file with the minimum necessary structure.
Set up this workflow

```
echo Hello, world!
```

First, let's name the workflow file. Change blank.yml to s3-depl. You can leave the filename as blank.yml, but it is best to give it a descriptive name.


s3-github-actions / .github / workflows / blank.yml Cancel
Start commit
Edit new file Preview
Spaces 2 No wrap
Marketplace Documentation
Search Marketplace for Actions

Copy and paste the code snippet into the editor. Copy and Paste, a developer's superpower 👨‍🔬 👩‍🔬.

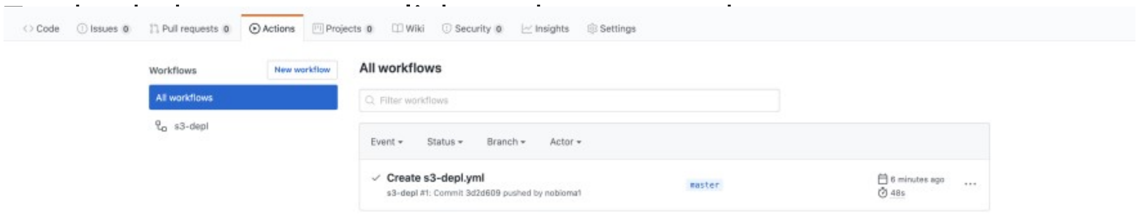
```
name: s3-depl

on:
  push:
    branches:
      - master

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ${{ secrets.AWS_REGION }}
      - name: Build React App
        run: npm install && npm run build
      - name: Deploy app build to S3 bucket
        run: aws s3 sync ./dist/ s3://<bucket-name> --delete
```

Mehhnnn! So many lines!!😨
Now, let's breakdown the code snippet above.

- name: We define the name of this action. This will be used to identify the action amid may others you may have.
- on: We define trigger with on : push also the branch. This workflow will run anytime you push code to the master branch.
- jobs: Workflow run is made up of one or more jobs and they run in parallel by default.
  - steps: A job contains a sequence of tasks called steps. Steps can run commands, run setup tasks, or run action in your repository and each step starts either with a uses: or a name:.
  - **actions/checkout@v2**: This action checks-out your repository, so your workflow can access it.
  - **aws-actions/configure-aws-credentials@v1**: This configures AWS credentials and region environment variables for use in other GitHub Actions.
  - **Build React App**: This step block installs the node packages and runs the build in the package.json file, which creates a dist folder in the root directory.
  - **Deploy app build to S3 bucket**: This deploys the newly created build to S3 bucket <bucket-name> (replace <bucket-name> with the name of your S3 bucket. Mine is s3-github-actions).

To save, click on the Start Commit then Commit New File. This will,

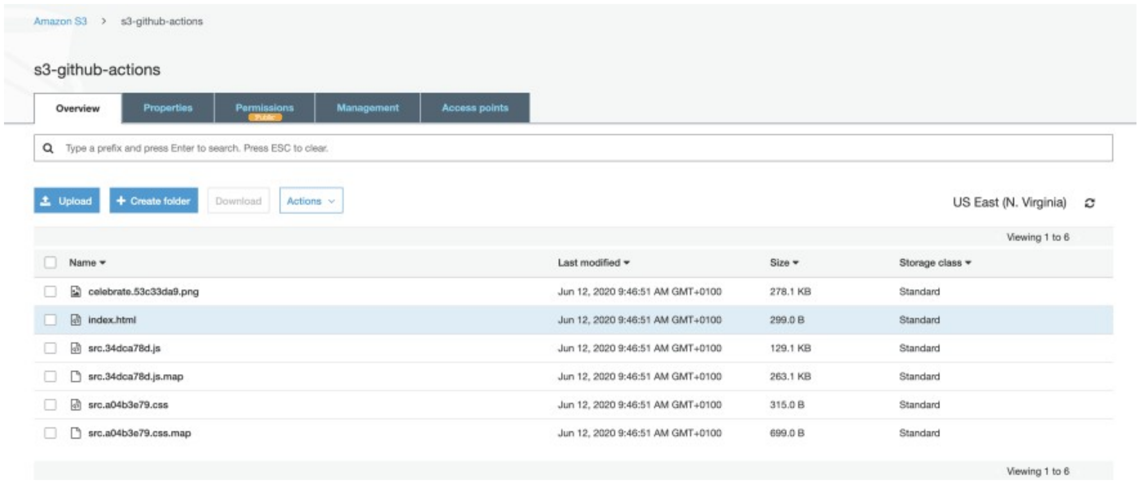
- save the action, creating a .github directory with a workflows directory in it that contains the new file s3-depl(the file name you used earlier)
- Trigger the action.



Create s3-depl.yml
s3-depl #1: Commit 3d2d609 pushed by nobioma1
master
6 minutes ago
48s

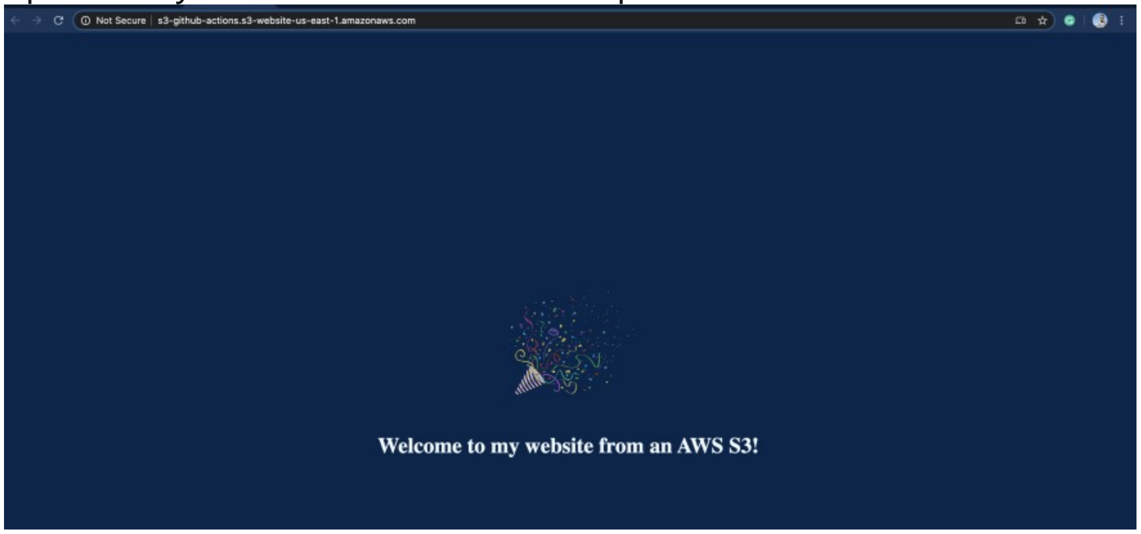
Voila!! The action ran successfully. Yay! Party After Party!! 🎊

You can now check your S3 bucket, you would see that the build files have been uploaded to it.



Name	Last modified	Size	Storage class
celebrate.53c33da9.png	Jun 12, 2020 9:46:51 AM GMT+0100	278.1 KB	Standard
index.html	Jun 12, 2020 9:46:51 AM GMT+0100	299.0 B	Standard
src.34dca78d.js	Jun 12, 2020 9:46:51 AM GMT+0100	129.1 KB	Standard
src.34dca78d.js.map	Jun 12, 2020 9:46:51 AM GMT+0100	263.1 KB	Standard
src.a04b3e79.css	Jun 12, 2020 9:46:51 AM GMT+0100	315.0 B	Standard
src.a04b3e79.css.map	Jun 12, 2020 9:46:51 AM GMT+0100	699.0 B	Standard

Our site is now live!!! On the browser, go to the `Endpoint` URL `(http://<s3-bucket>.s3-website-<s3-region>.amazonaws.com)` that we came across when enabling `Static Website Hosting`. Now, any change you make to your react app will build and upload to your S3 bucket which will update live.


Not Secure | s3-github-actions.s3-website-us-east-1.amazonaws.com

Welcome to my website from an AWS S3!

You can go on to work with Github Actions by triggering an action on Pull Request that might run some CI tests and perform several steps before deploying to your S3.