Use Terraform to create s3 bucket on AWS

Use GitHub actions to upload files on AWS s3 bucket created above.

Store all your code in your personal GitHub account. Demo after all done!

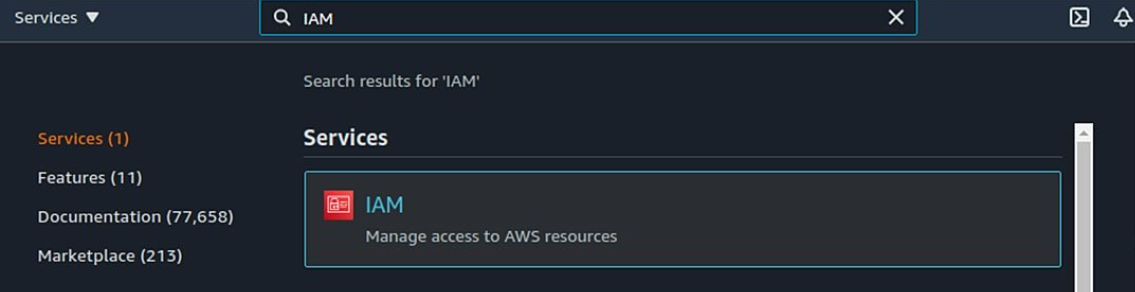
Prerequisites

1. An AWS account
2. An AWS IAM role with S3 Permissions
3. Access key ID and Secret of that account
4. GitHub account

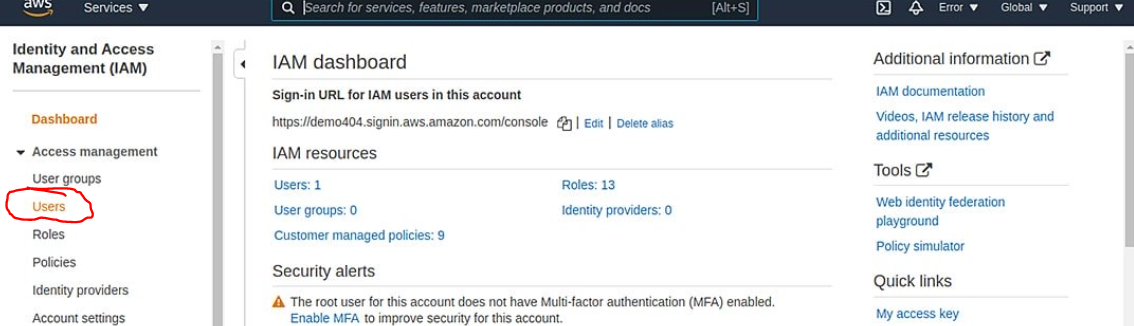
**How to create an Access Key and Secret Key?**

1. Login to your [***AWS Account***](https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Faws.amazon.com%2Fmarketplace%2Fmanagement%2Fsignin%3Fstate%3DhashArgs%2523%26isauthcode%3Dtrue&client_id=arn%3Aaws%3Aiam%3A%3A015428540659%3Auser%2Faws-mp-seller-management-portal&forceMobileApp=0&code_challenge=QCY-5n8K1nGf8LcWsSGfJBD83Phkwx19Myi1jJ0mqH4&code_challenge_method=SHA-256)
2. 2. Select **IAM**

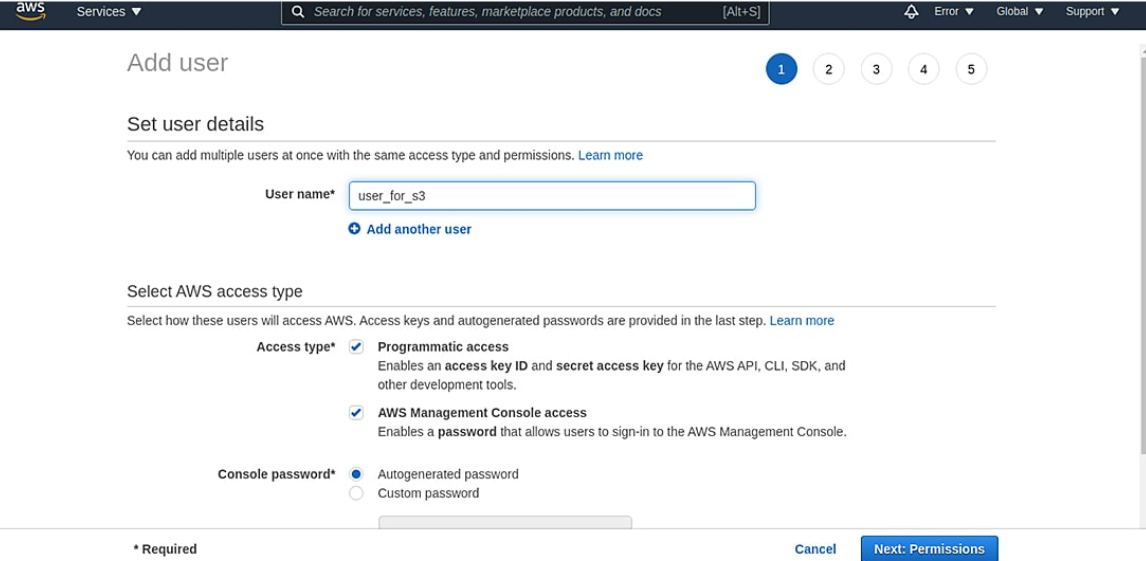
In Management Console, search for **IAM**.



3. On the left side of the panel, select ***User***.



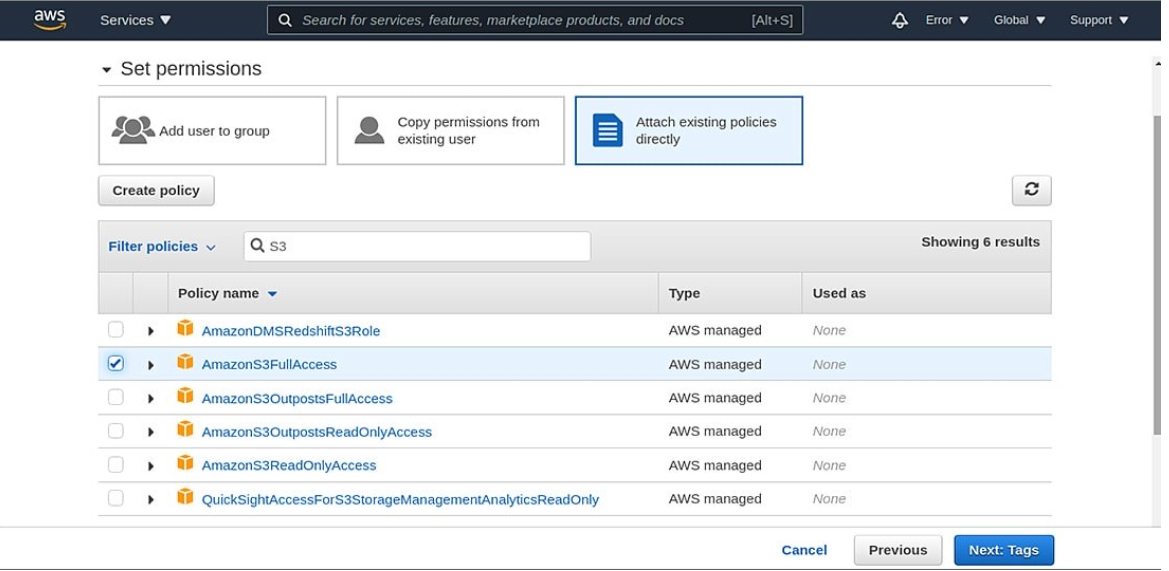
1. Select ***Add Users*** and enter details.



Please keep in mind to select **Programmatic access** in Access type to get Access Key ID and Secret Key.

1. Attach policy

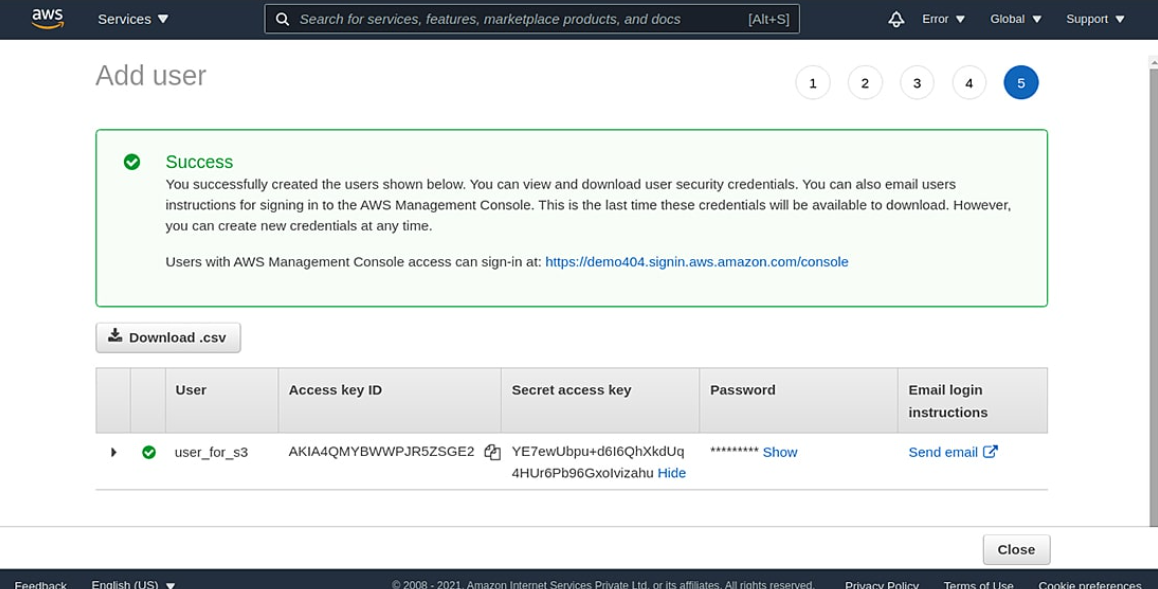
After entering the details, attach a policy for S3 as shown below



Select **Next: Tags** button displayed below and then **Add Tags** (optional)

Review your user policy and then **Create User**.

After Successful IAM user creation, you will see a message (as shown below) with your Access key and Secret key



**Note- Store this Access Key and Secret Key to your Local Computer as AWS does not allow you to retrieve secret keys after its creation.**

Now, moving towards how to create an S3 bucket using Terraform.

## Steps to create an S3 bucket using Terraform

**variables.tf**

This is the place where we will store all the AWS secrets such as Access Key ID , Secret Key, Region.

***aws\_access\_key*** – It makes an API call to AWS resources from your machine.

***aws\_secret\_key*** – Secret Access Key that’s associated with Access Key.

***aws\_region*** – The AWS region where you want to create all your resources.

**Providers.tf**

## In the provider.tf file , We will mention the provider as AWS and the region where the S3 bucket should be created.

## 

And the creds.tf file. While holds the AWS credentials and let the terraform to create the S3 bucket.

You can also configure AWS profile to access the credentials instead of directly using the credentials in creds.tf file.

## 

## The below script will create one s3 bucket , The ACL of the bucket will be Private and with the versioning enabled

## 

## Run terraform Init to initialize the folder you are working on

## Run **terraform plan** to verify the script. It will let us know what will happen if the above script is executed

## Now run **terraform apply** to create s3 bucket.

## Setup GitHub Account

## Create new repository

## 

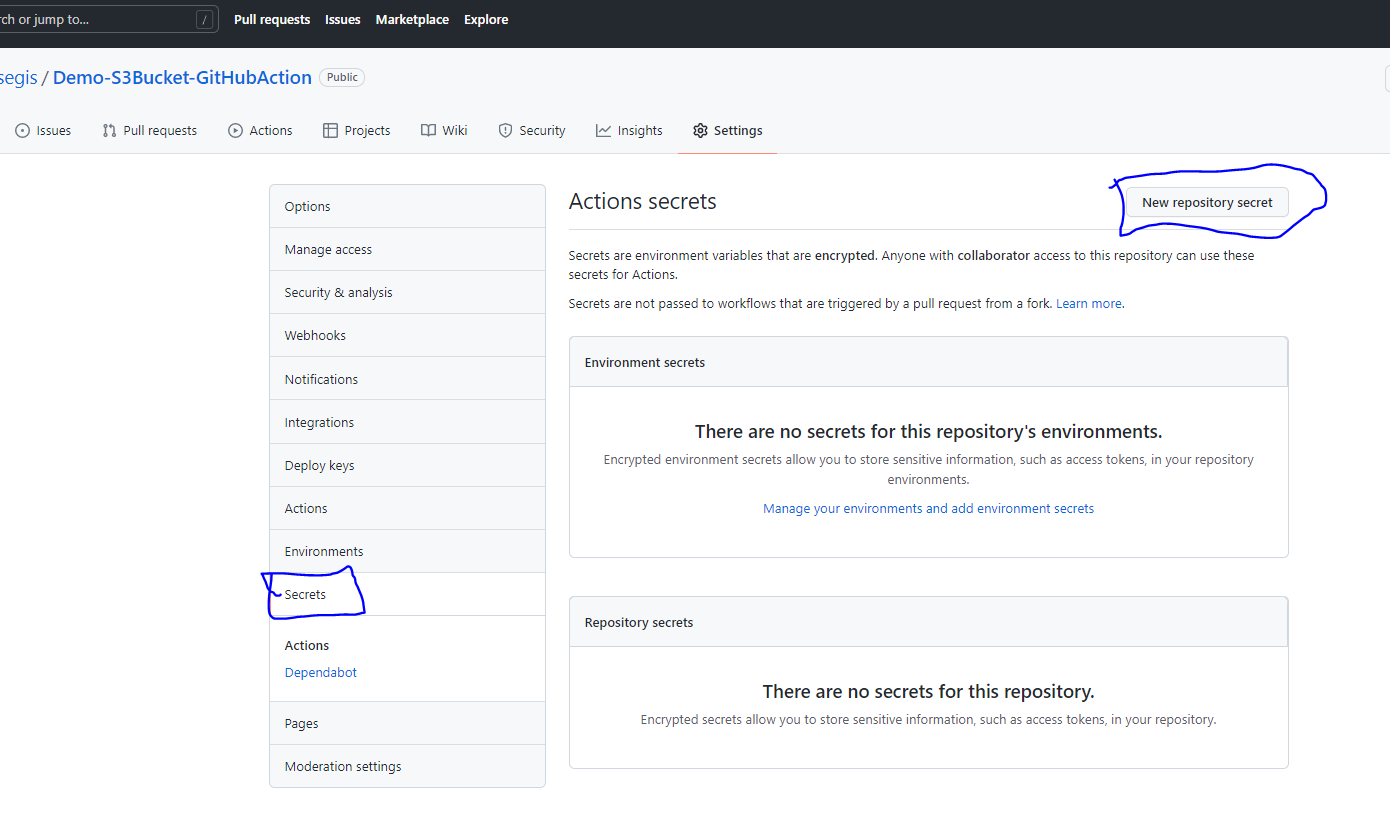


Just before you push the code up, you will need to add the secrets. This involves going to the Settings of the repository. On the left, you will see "Secrets". Add the secrets:

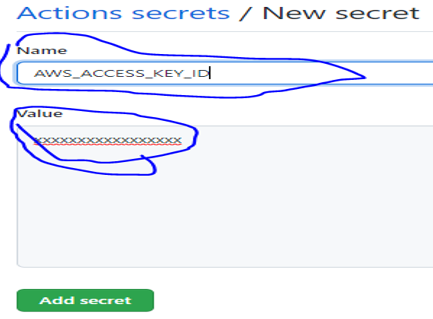
* AWS\_Bucket\_Name
* AWS\_Access\_Key
* AWS\_Access\_Secret

## 

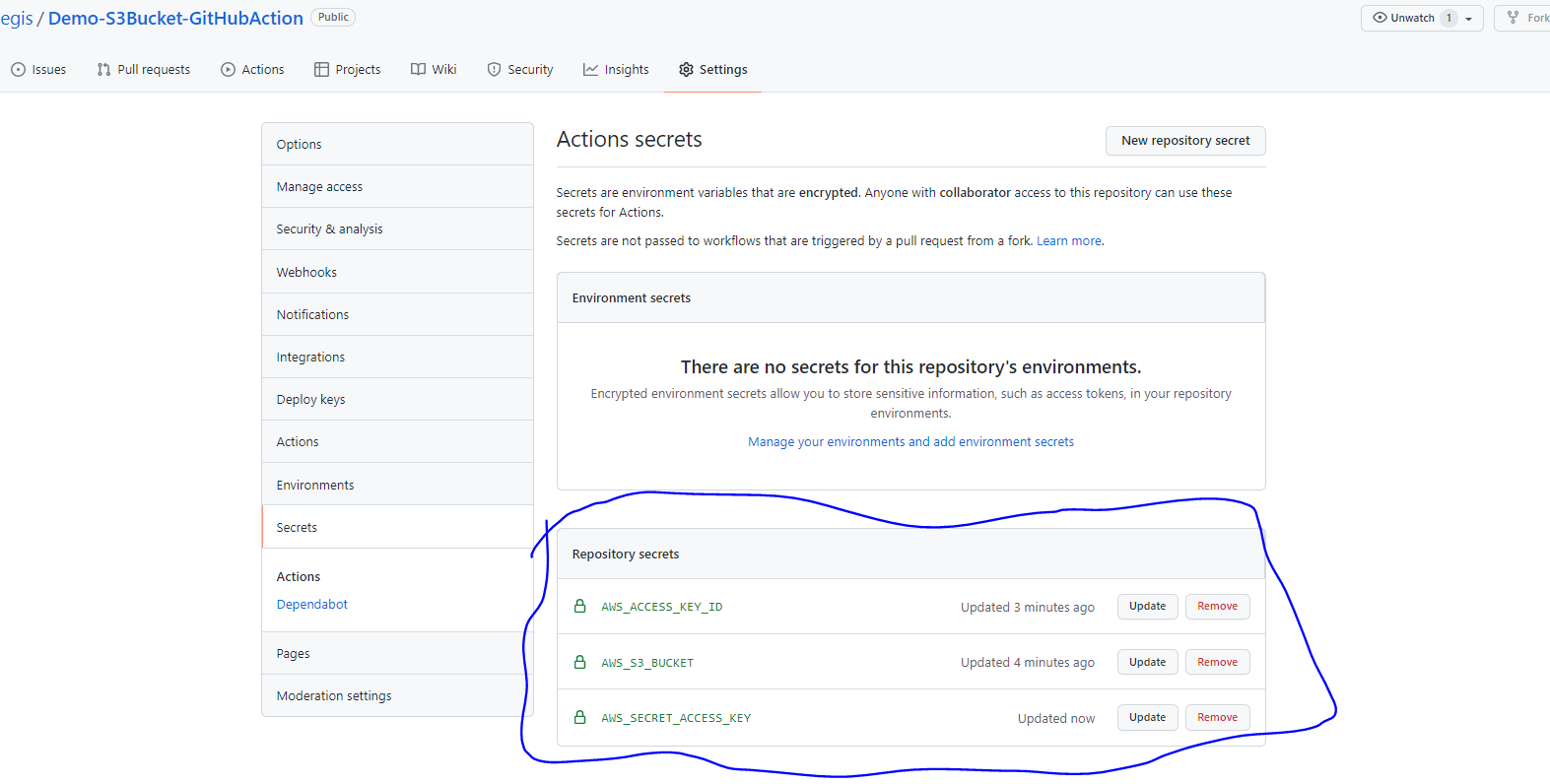
## Next, select Secret and then New repository secret to store your AWS\_Bucket\_Name you have created earlier then add secret. Proceed the same with AWS\_Access\_Key and AWS\_Access\_Secret your have downloaded and save from your AWS account.



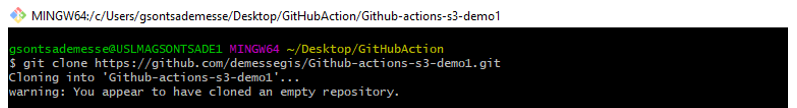
Ensure the Name matches with what you have on your. github/worklows/file



See below your repository secret



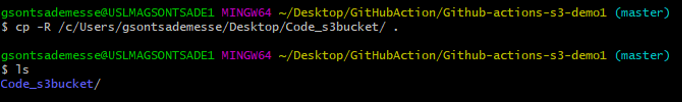
Then clone your GitHub repository on any command line you would like to “ git bash, Visio studio, ..etc)



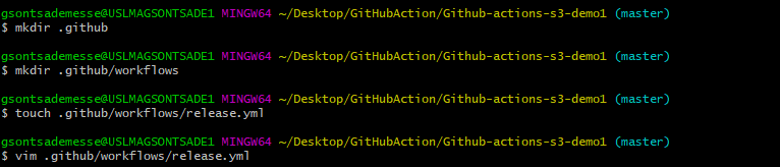
Then, get into the repository you just created



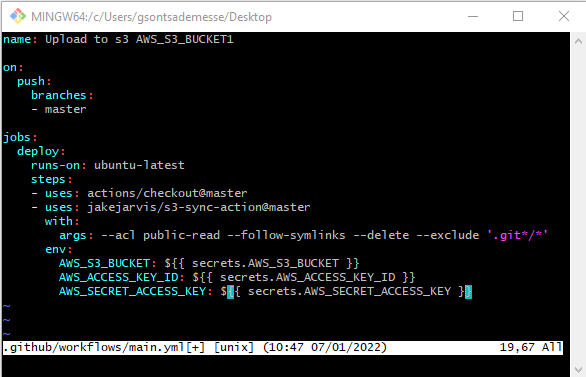
Next, copy the file you would like to store into your S3 bucket,



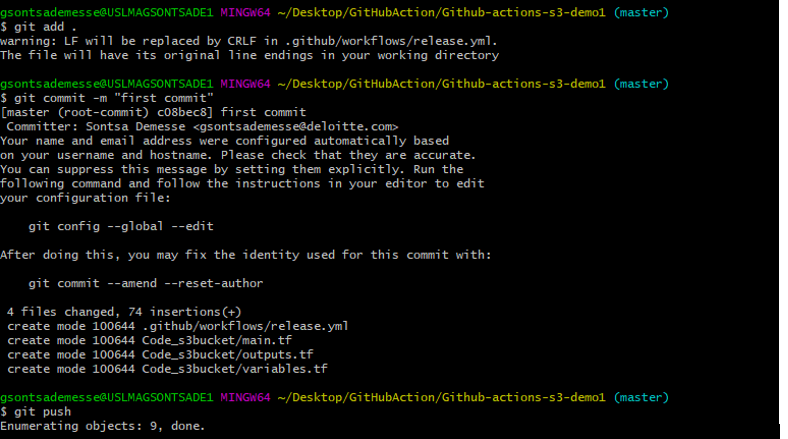
Next create a .github repository, then another workflows repository into .github, and a files into workflow repository. Then create a file to paste your deployment code



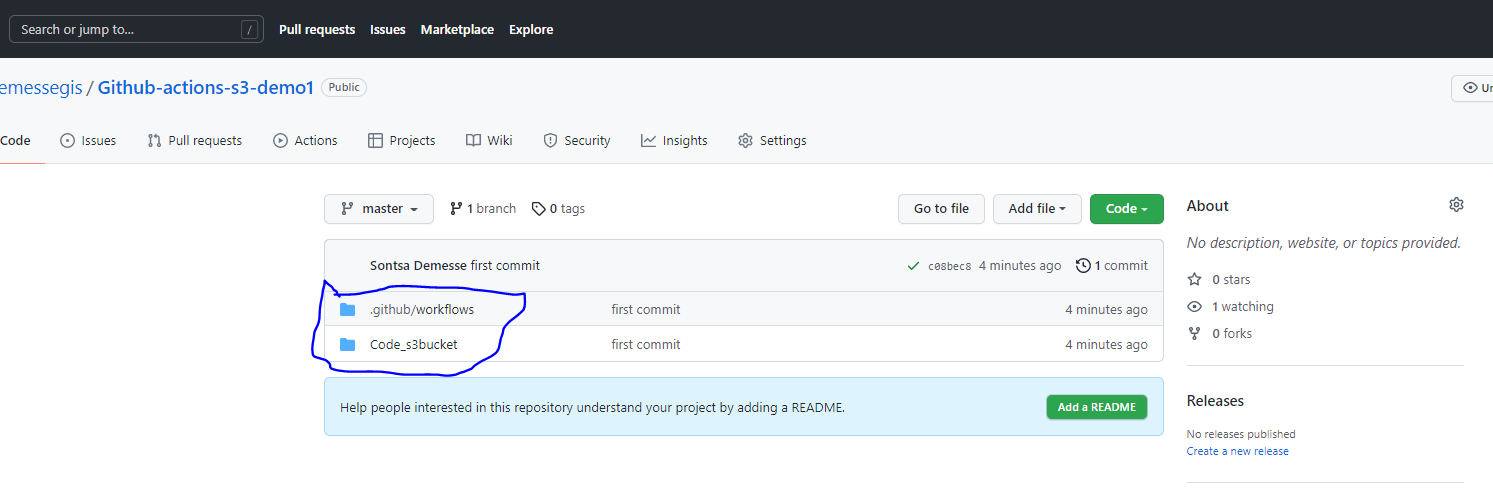
Here is your deployment code to be uploaded



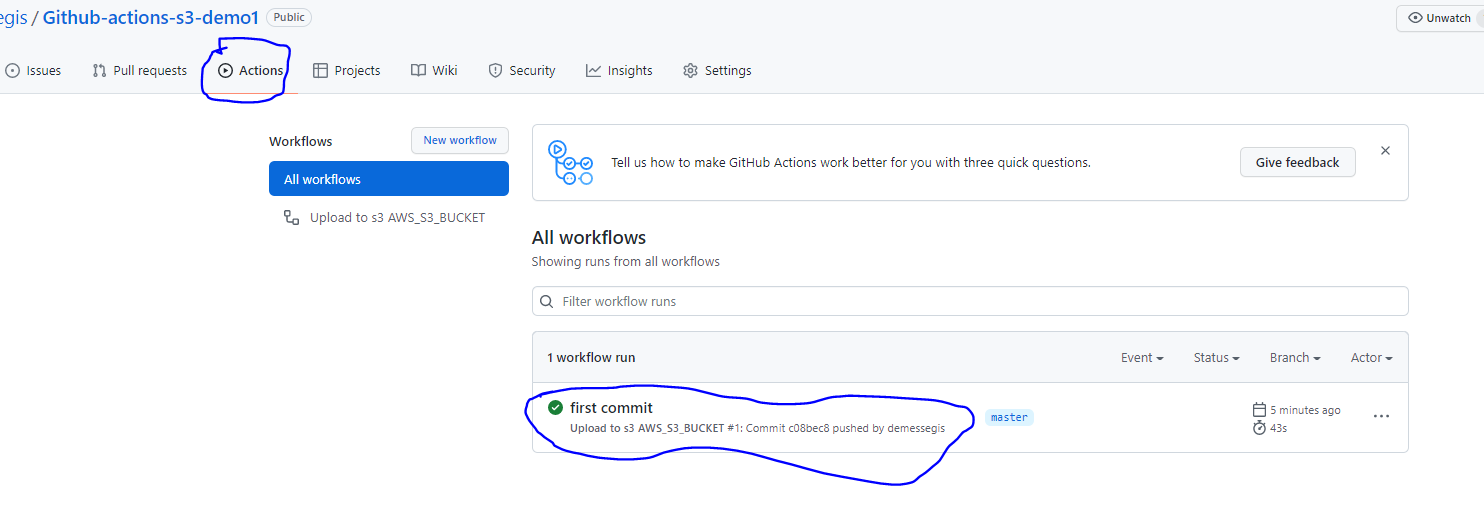
Next add, commit and push your file into your github.



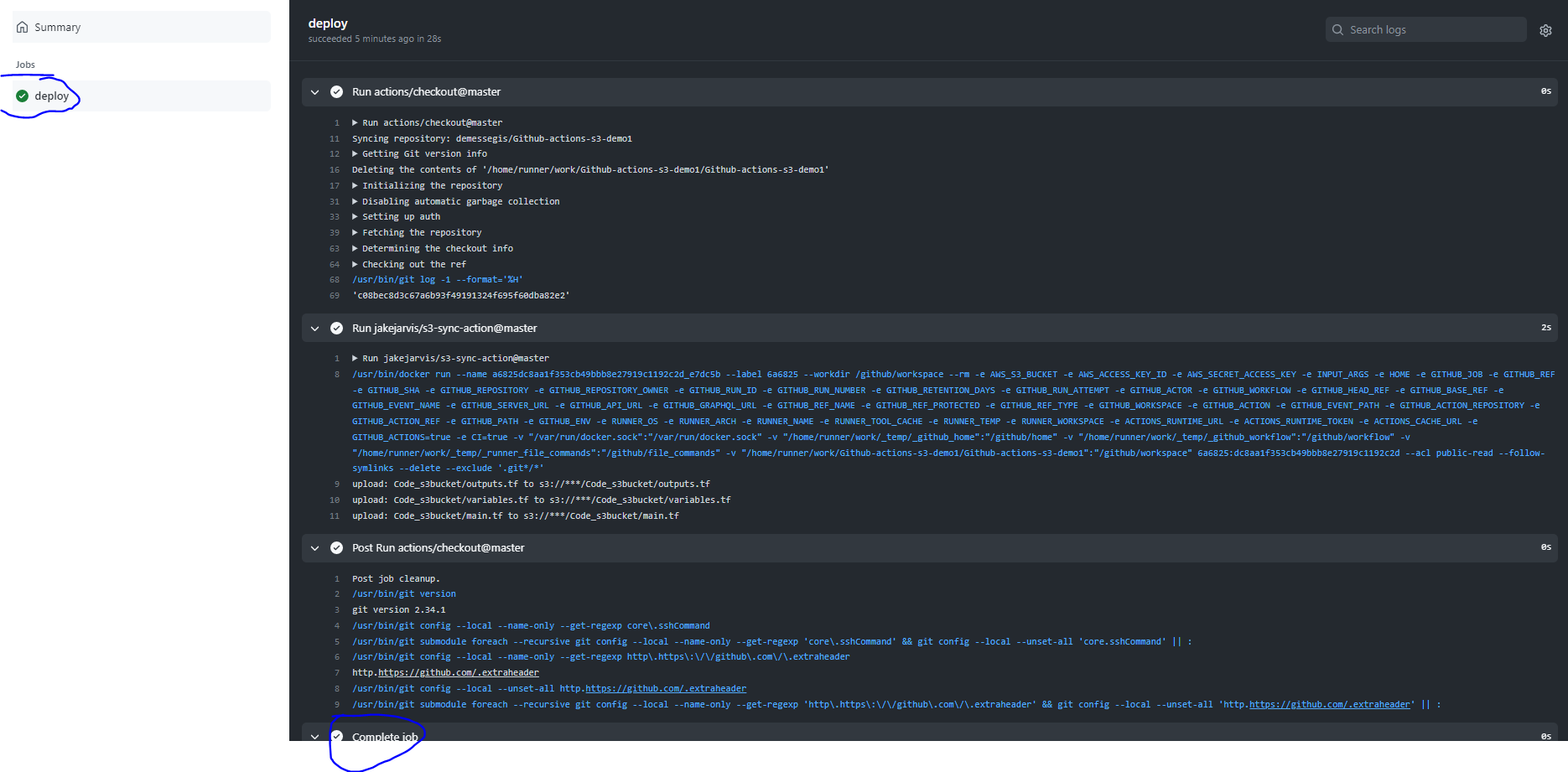
Here you can see you file uploaded into your github account



Then click on actions to confirm your deployment file



Next, confirm your job is successfully completed



Next, go to your AWS account and S3 bucket to confirm your folder is successfully uploaded into your s3 bucket.

