Github Actions

Prerequisite - Go through the concepts of Git and Github

<https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>

Link: <https://www.youtube.com/watch?v=-U-eUHI6euM&list=PLhW3qG5bs-L8OlICbNX9u4MZ3rAt5c5GG>

Setting up SSH key for gitbash to communicate with your git repository :

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Have – gitbash, terraform, aws cli (optional), vscode and git installed.

GitHub Actions with terraform and aws

Aws related action : <https://github.com/aws-actions>

Checkout action : <https://github.com/actions/checkout>

Terraform Checkout : <https://github.com/hashicorp/setup-terraform>

Authentication

A diagram of a computer

Description automatically generated

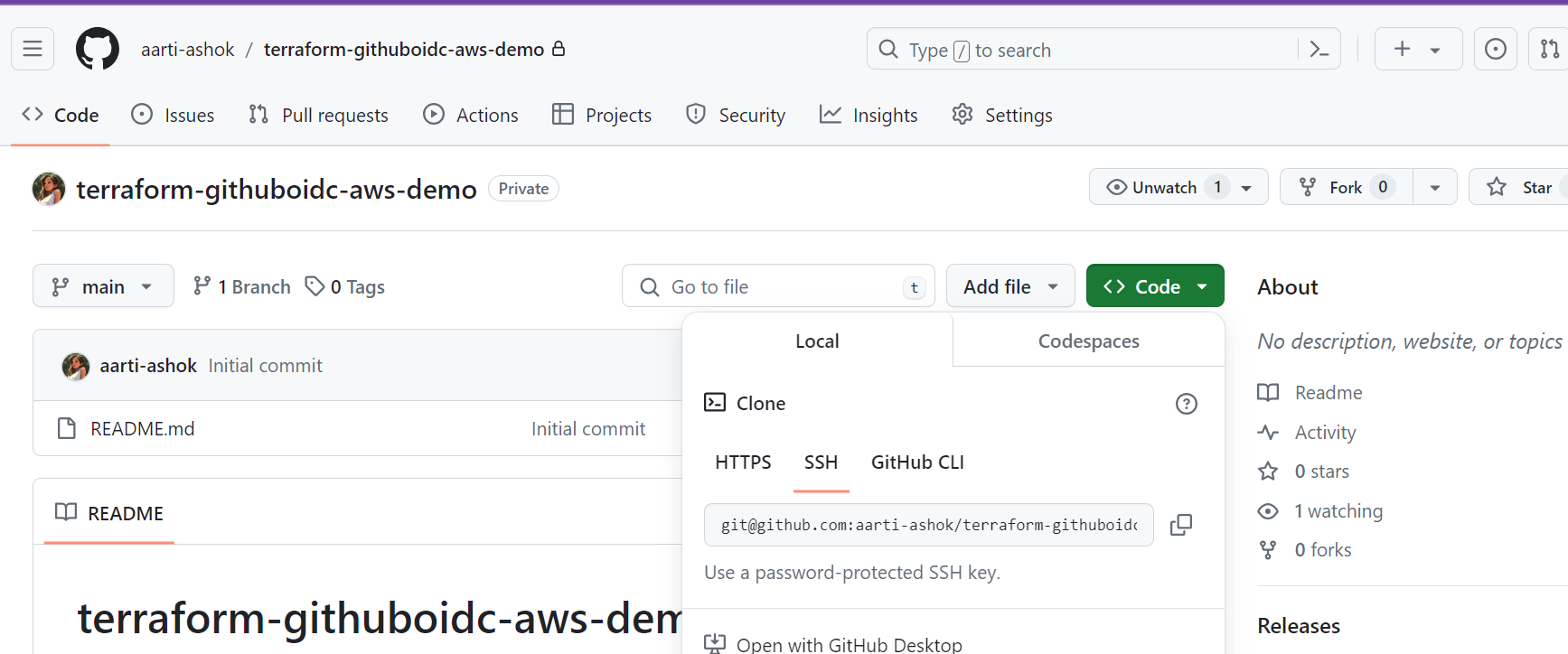
Steps in Details

1. Create a **Private** Repository

A screenshot of a computer

Description automatically generated

1. Copy the code and clone it in local.

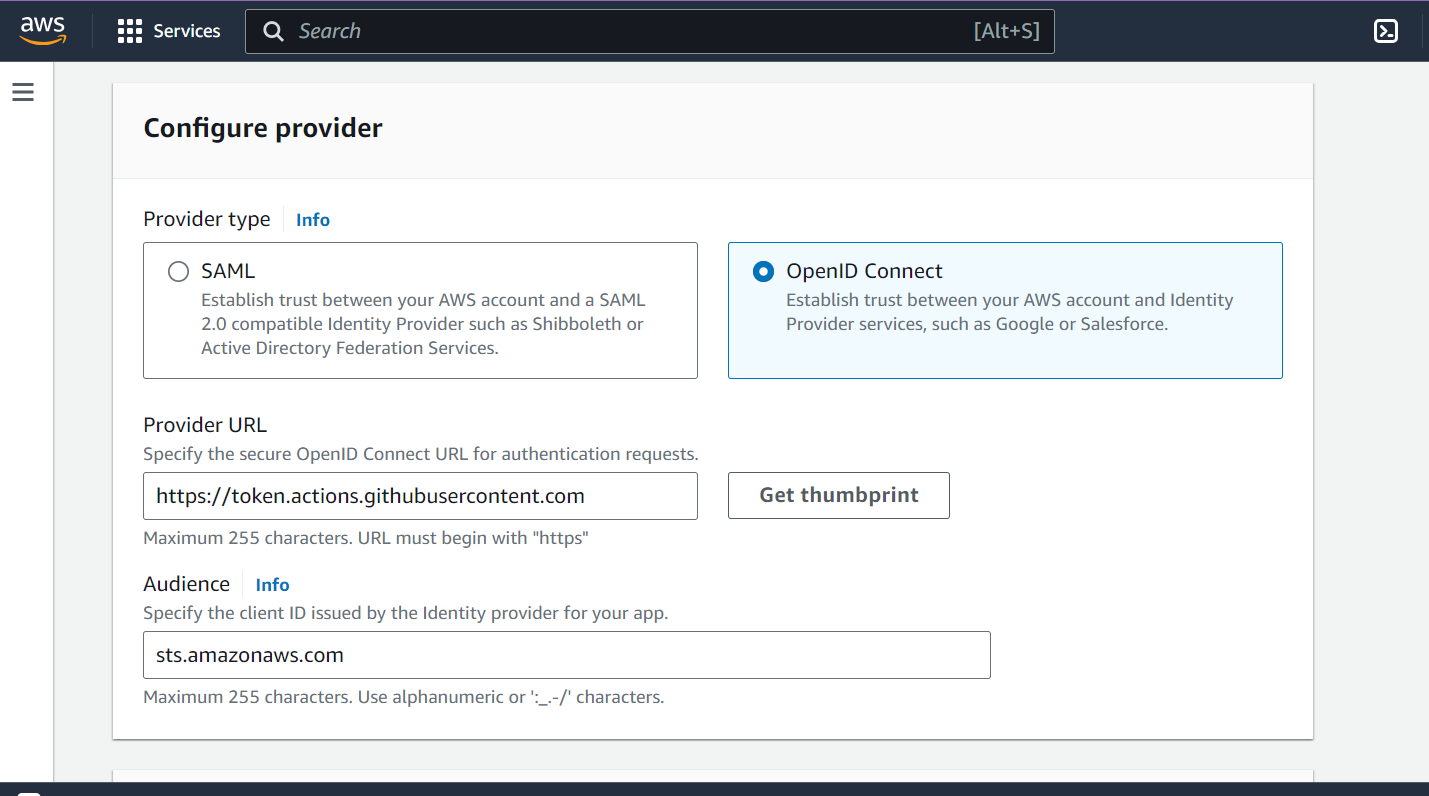


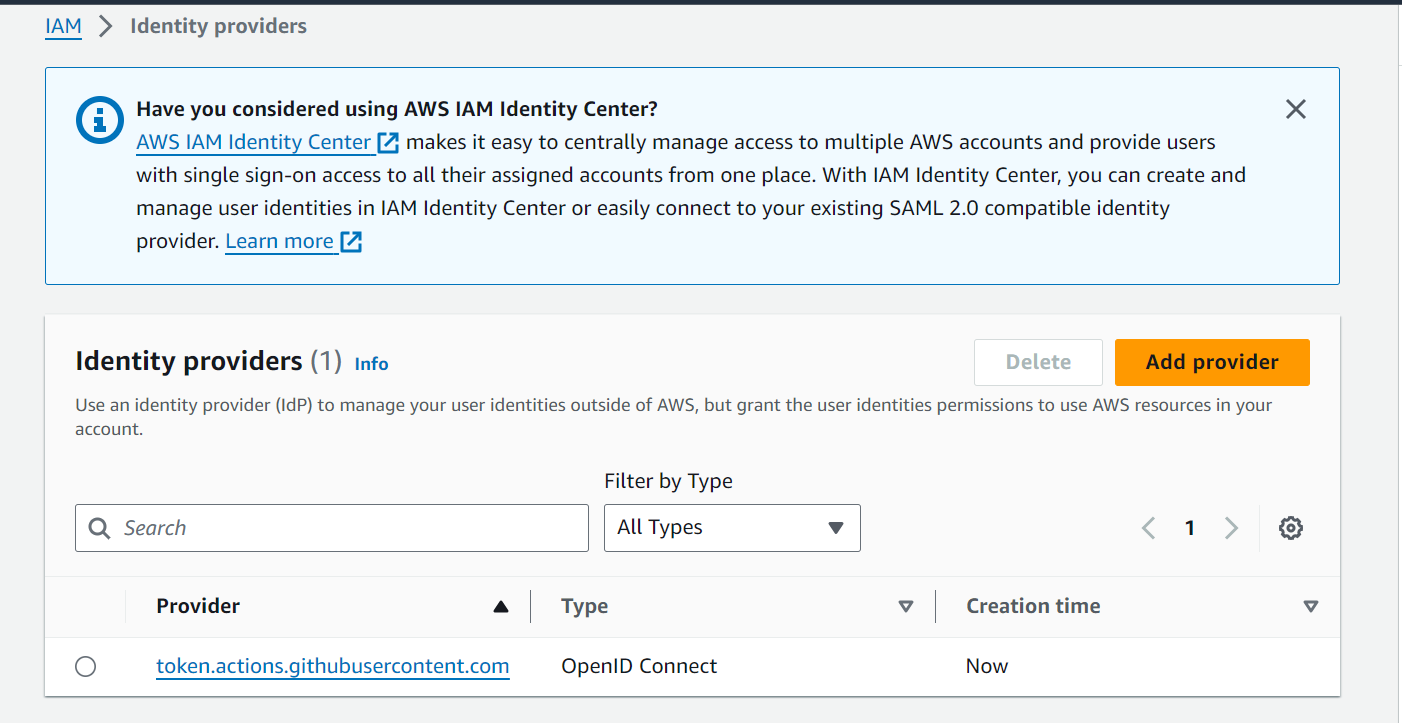
A screenshot of a computer program

Description automatically generated

1. Setting up Github OIDC for github workflows

Open up AWS console and go to IAM ->Identity Provider and create Provider->select OIDC and fill the details as per screenshot below and after that click on Get thumbprint, Add the provider





After adding the provider, you will give necessary permissions to run terraform so we will create s3 bucket to store terraform state file and then give terraform other roles to create infrastructure

Now we need roles for the github to access aws

Go to IAM - >create role->custom policy (add your aws account and repo name)

{

"Version": "2008-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Federated": "arn:aws:iam::YOUR\_ACCOUNT\_NUMBER:oidc-provider/token.actions.githubusercontent.com"

},

"Action": "sts:AssumeRoleWithWebIdentity",

"Condition": {

"StringLike": {

"token.actions.githubusercontent.com:sub": "repo:YOUR\_GITHUB\_USERNAME/YOUR\_REPO\_NAME:\*"

}

}

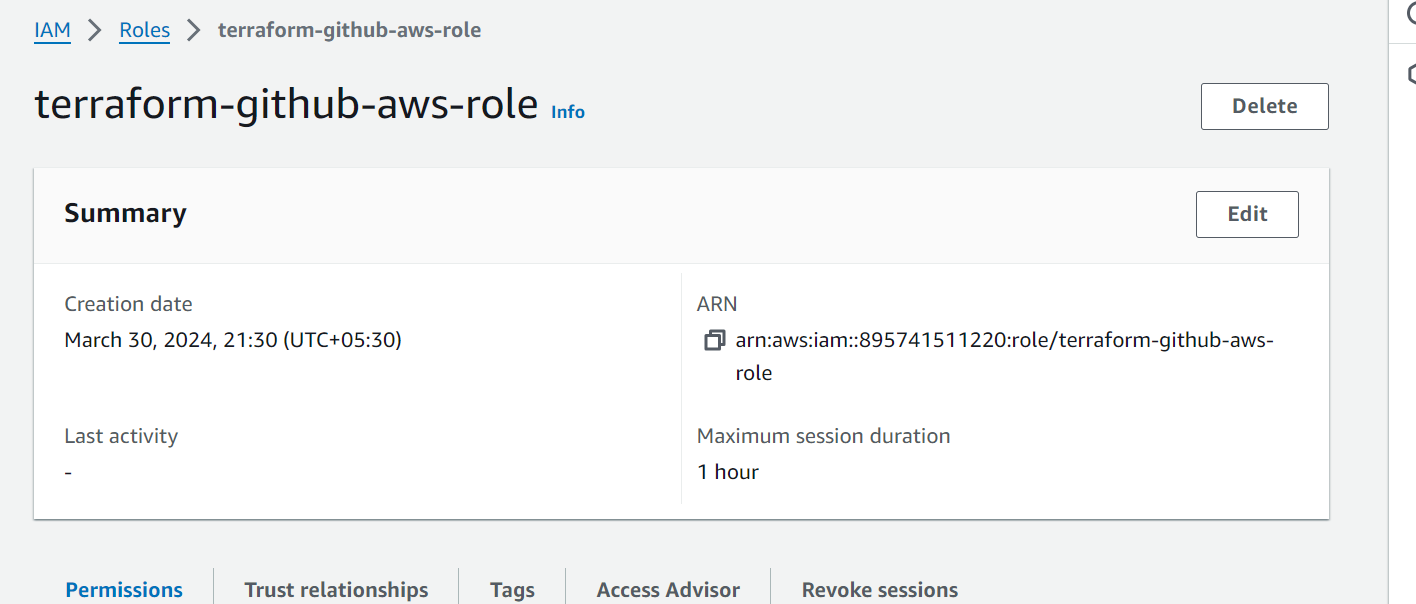
}

]

}

Once role is created add required policy i.e. what all actions you need like s3, network, ec2.. For demo purpose I have added admin access

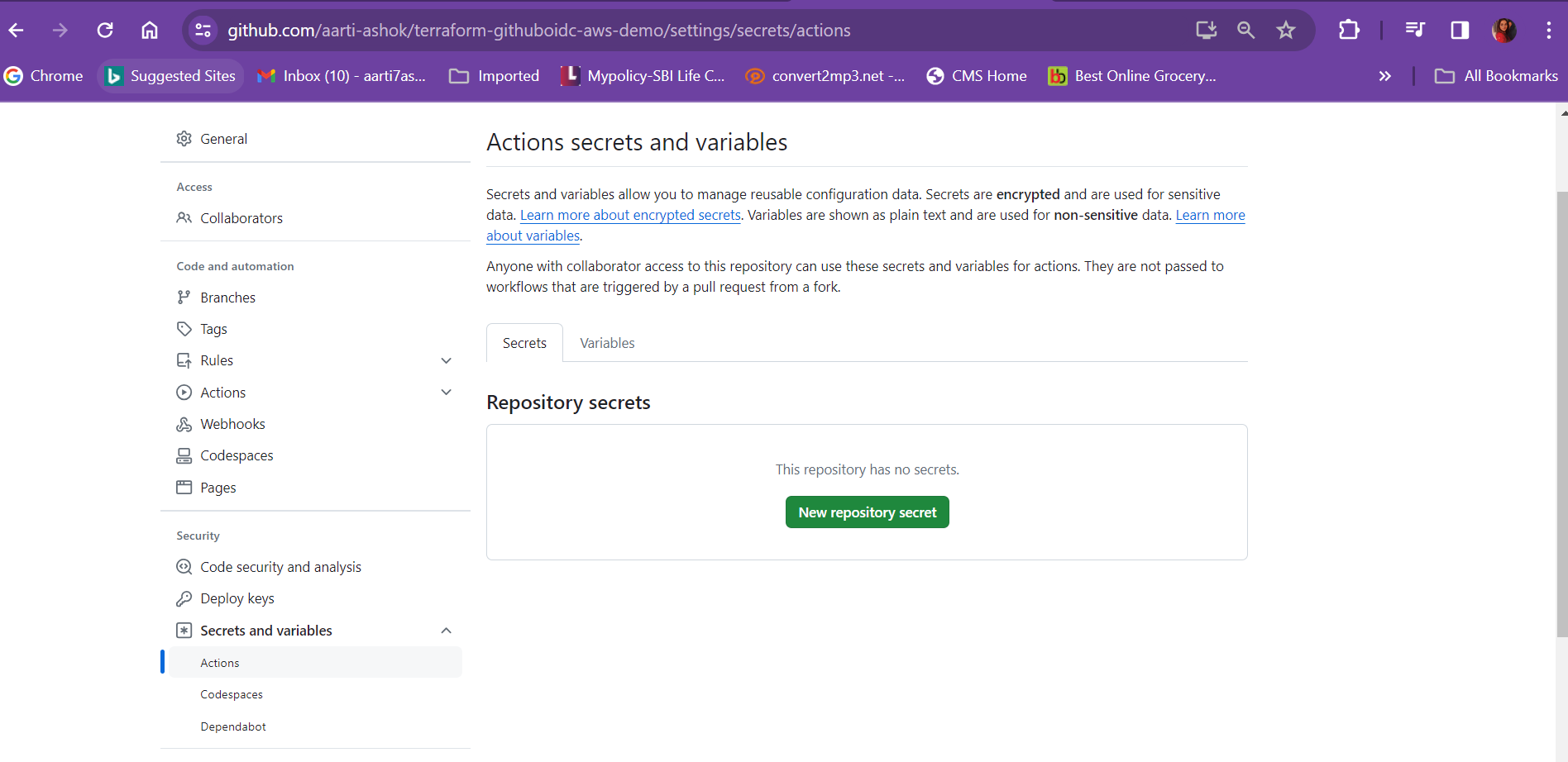
Note the ARN of the role, we will use this in github so that we can run workflows



Now let’s use the terraform configuration files we learnt earlier and deploy. Also download a gitignore file for terraform and include in your repo.

Now we need to add some secrets in github as well so that github can contact aws to run the terraform commands as part of workflow

Go to Github- Settings->Secrets & Variables ->Actions.



Add below secrets like below.

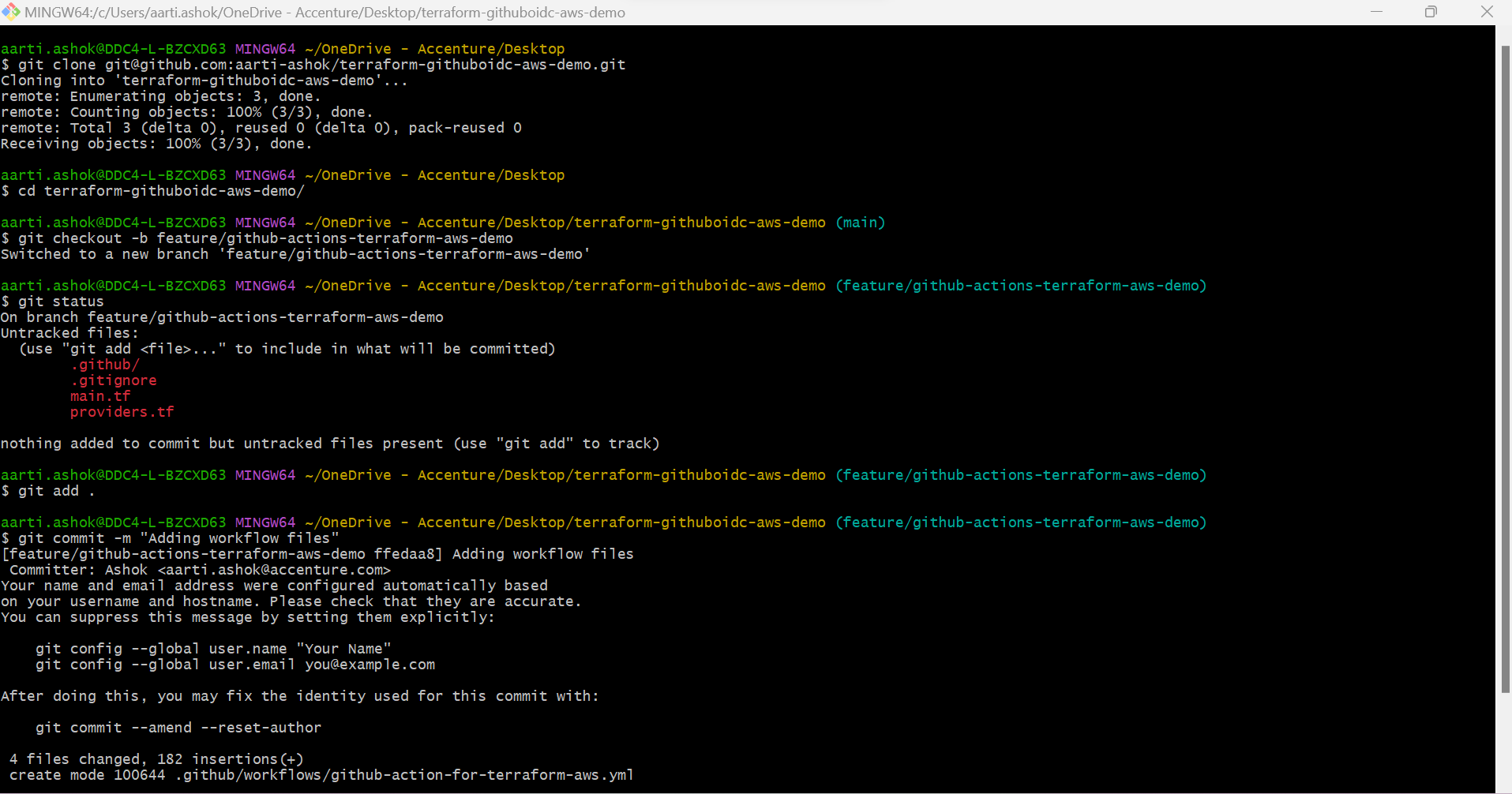
A screenshot of a computer

Description automatically generated

Details of Github Actions is given in below link:

<https://github.com/orgs/actions/repositories>?

Lets run the workflow file:



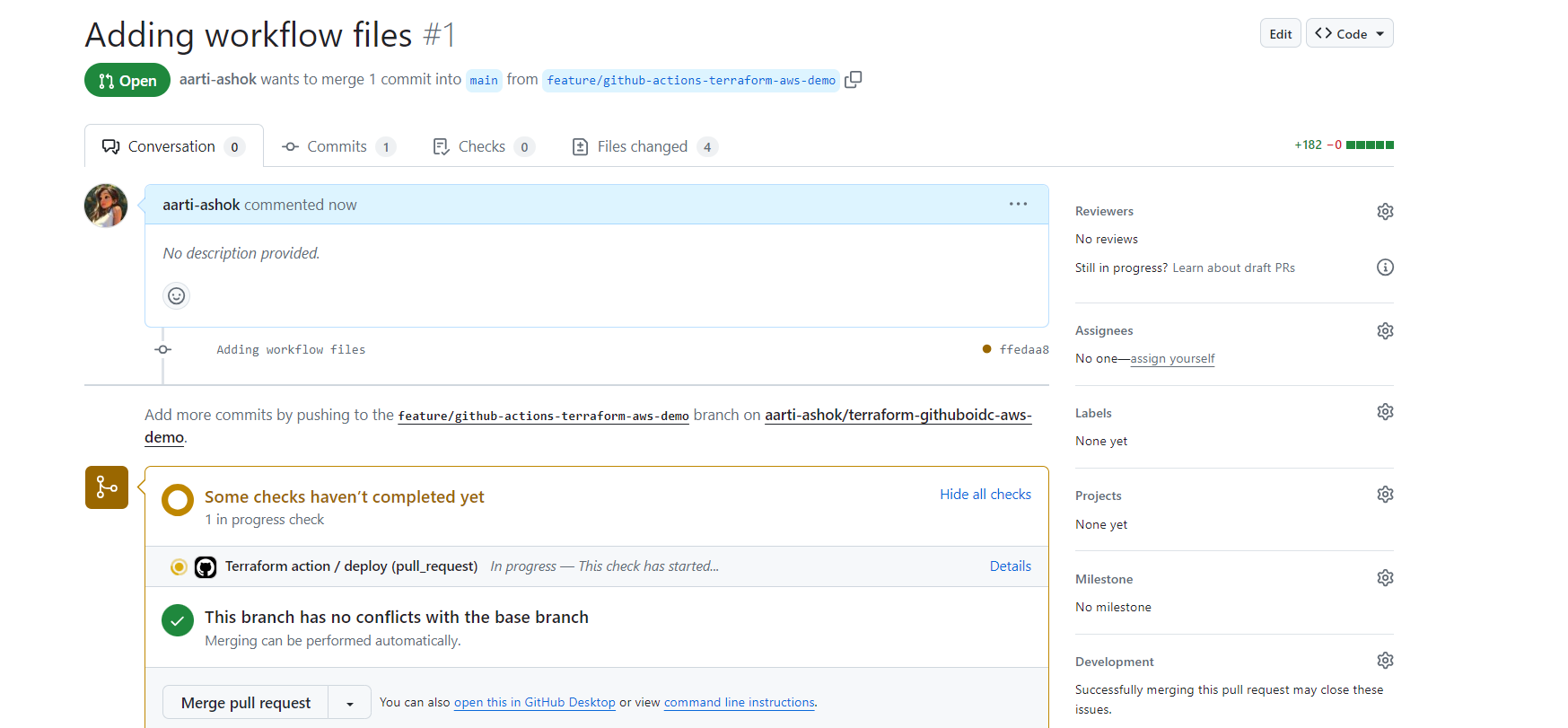
A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Now click on Compare and Pull Request



We will see github actions running, go to actions

A screenshot of a computer

Description automatically generated