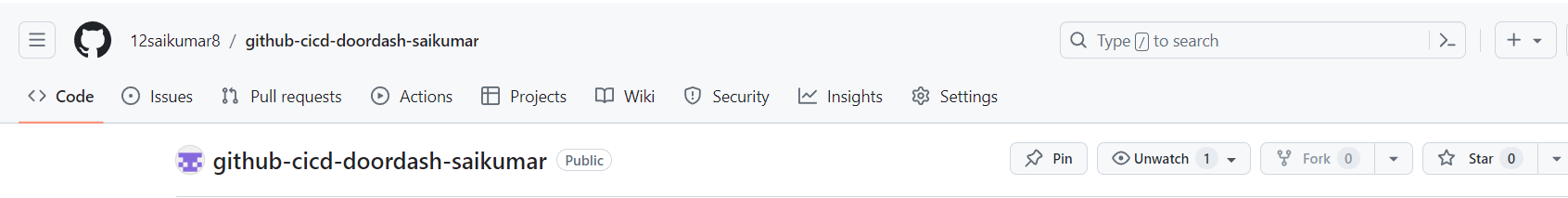
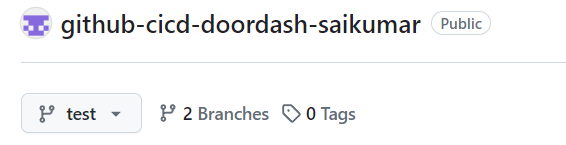
**Processing for Door Dash Delivery Data Overview:**

This assignment involves creating an automated AWS-based solution for processing daily delivery data from Door Dash. JSON files containing delivery records will be uploaded to an Amazon S3 bucket. An AWS Lambda function, triggered by the file upload, will filter the records based on delivery status and save the filtered data to another S3 bucket. Notifications regarding the processing outcome will be sent via Amazon SNS

1. First we will create a repository in Github



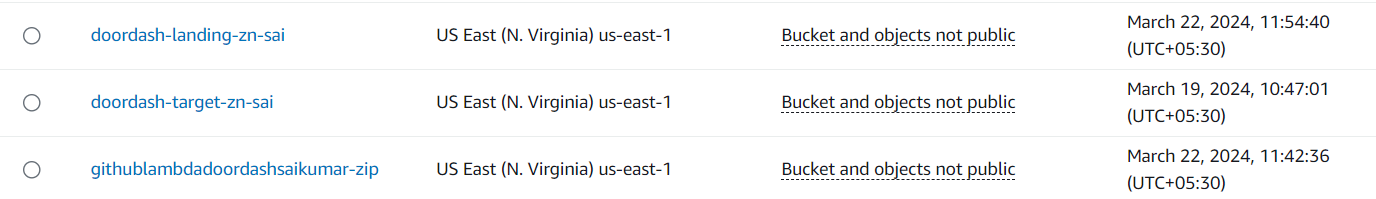
Next will add a new branch to it 🡪 git checkout -b test

  
 the branch will be created in it.

2).Next

Will create 2 buckets in s3 🡪 they are i).doordash-landing-zn-sai for daily file uploading here the daily file is uploaded and ii). Doordash-target-zn-sai this bucket is target bucket which receives processed file.

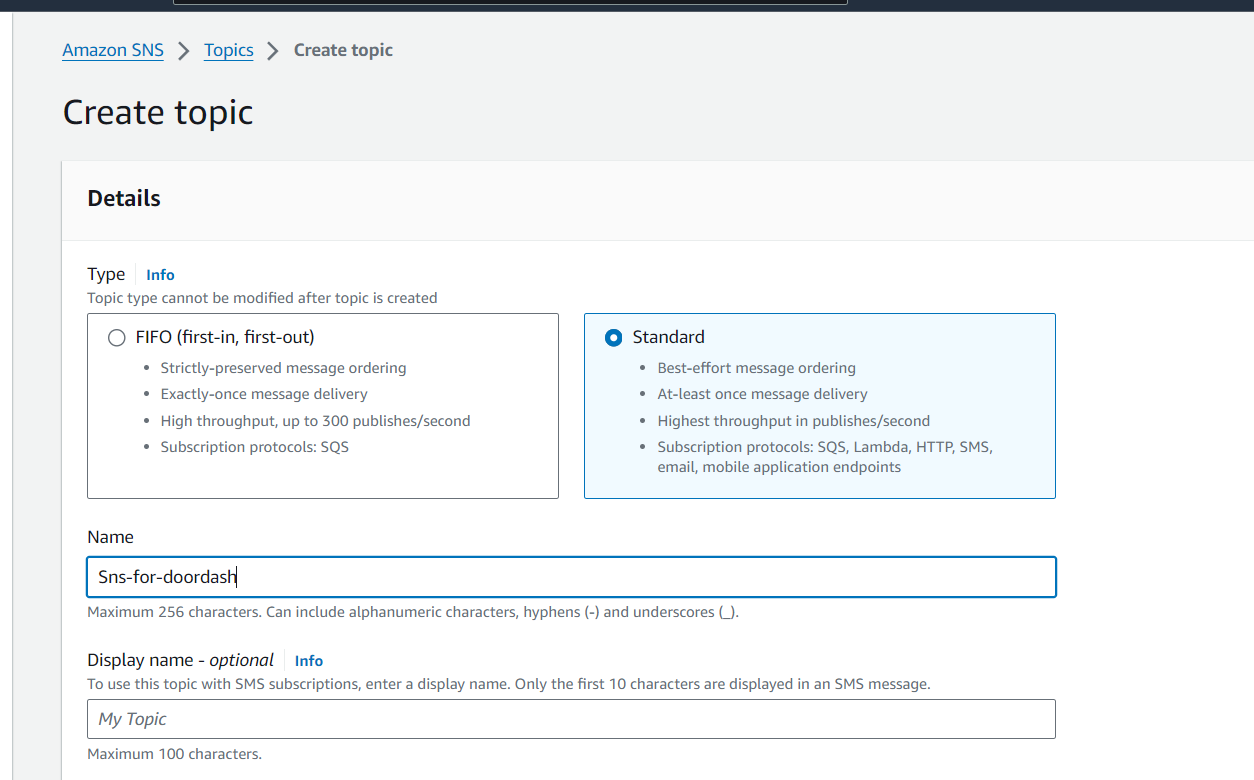
And the githublambdadoordashsaikumar-zip bucket is for storing the deployment\_package.zip



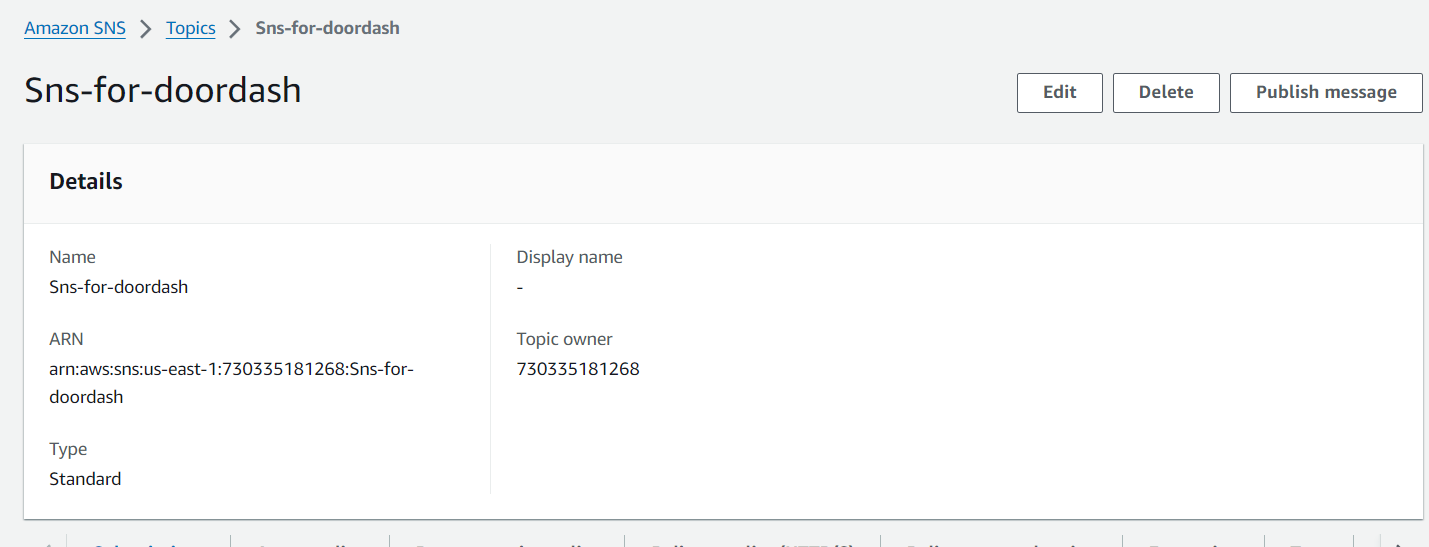
● Set Up Amazon SNS Topic: ○ Create an SNS topic for sending processing notifications.

○ Subscribe an email to the topic for receiving notifications

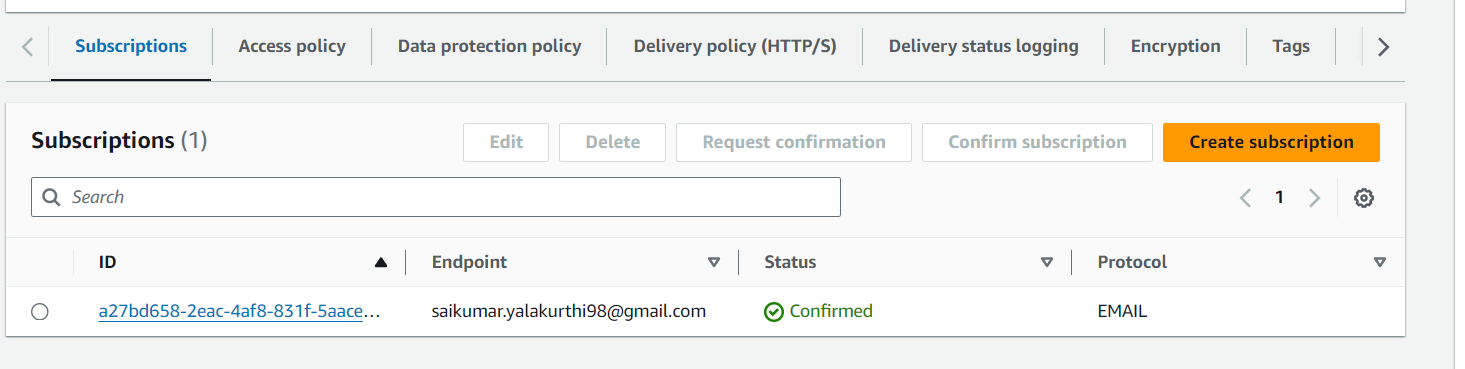
3). next we will setup SNS(simple notification service) and subscribe the mail



It will create like below then



Click on create subscription and create to subscribe our mail for receiving notifications.

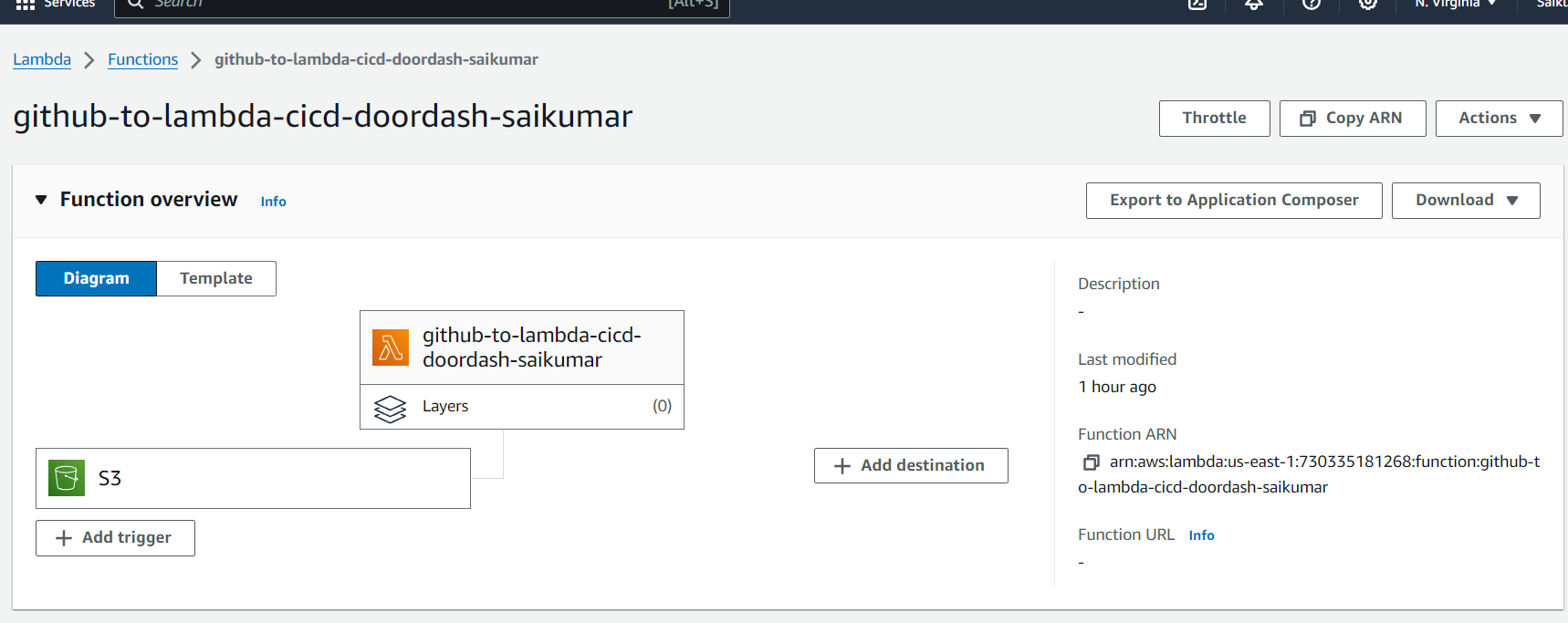


And confirm by acceptping it on your mail

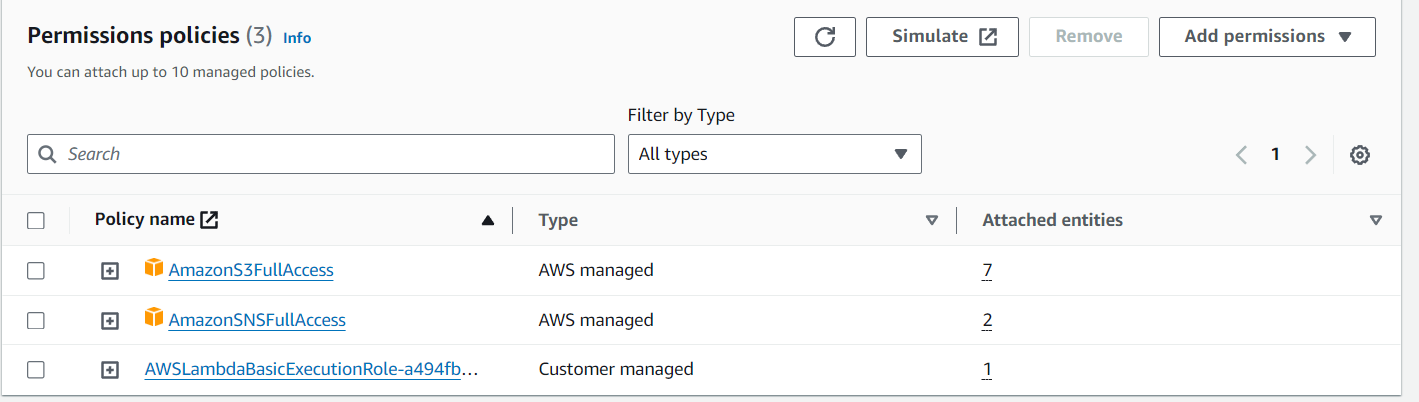
4).Then next

* ● Create and Configure AWS Lambda Function:
* ○ Create a Lambda function using Python runtime.
* ○ Add the pandas library to the function's deployment package or use a Lambda Layer for pandas.
* ○ Use the S3 trigger to invoke the function upon file uploads to doordash-landing-zn.

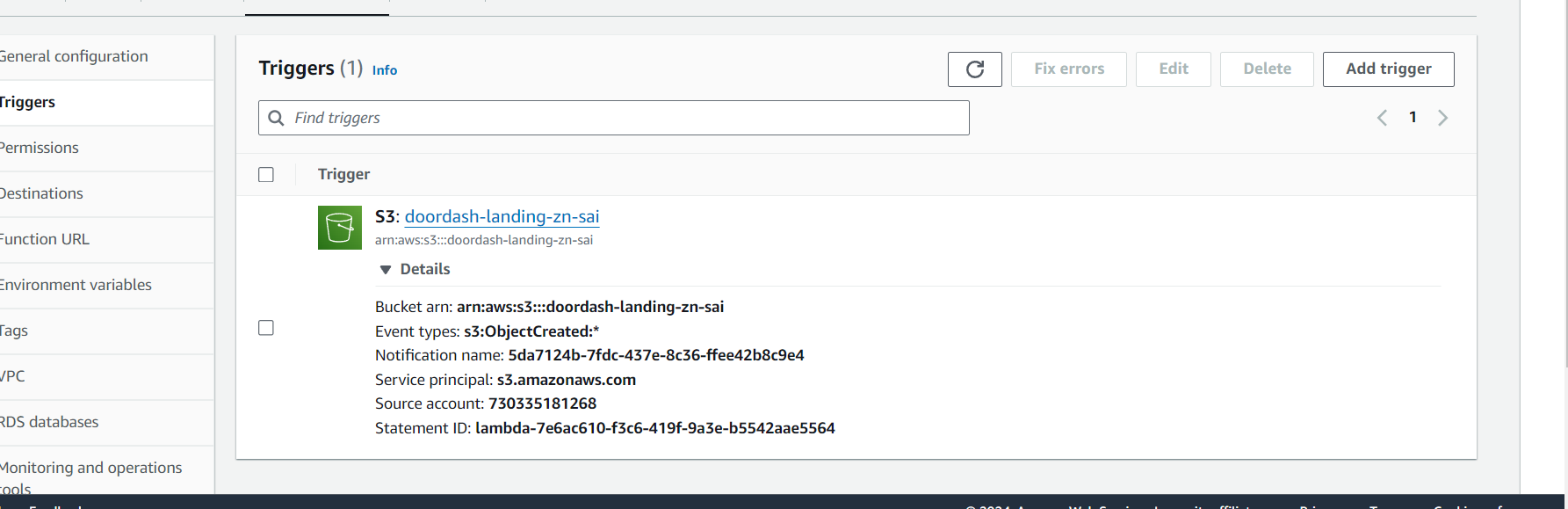
Next will create lambda\_function (github-to-lambda-cicd-doordash-saikumar) and S3 bucket as trigger below



Below we will will add s3full access and sns full access permissions for this lambda function



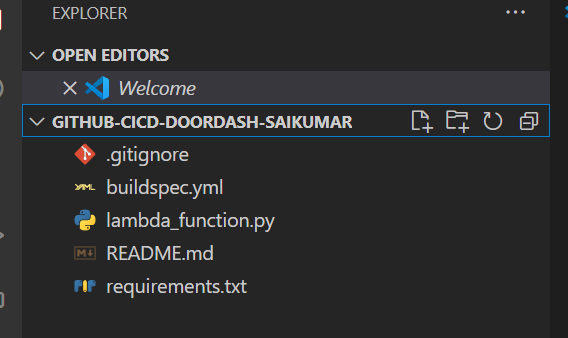
And then the trigger .The below are bucket name and details that added as trigger



5).Next will create lambda\_function in Vscode as per below requirements i.e 🡪{

* ○ The Lambda function should:
* ■ Read the JSON file into a pandas DataFrame.
* ■ Filter records where status is "delivered".
* ■ Write the filtered DataFrame to a new JSON file in doordash-target-zn using the specified format.
* ■ Publish a success or failure message to the SNS topic }🡨

And also we add SNS topic in lambda code to receive success or failure messages  
  
Below we will create directory(GITHYB-CICD-DOORDASH-SAIKUMAR) first and we will add all the files lambda\_function.py file , requirements file etc.. in that



The lambda function code is below

import boto3

import pandas as pd

import json

# Initialize S3 and SNS clients

s3\_client = boto3.client('s3')

sns\_client = boto3.client('sns')

# Define the SNS topic ARN

sns\_topic\_arn = 'arn:aws:sns:us-east-1:730335181268:Sns-for-doordash'

def lambda\_handler(event, context):

    try:

        print(event)

        # Get the S3 bucket and object key from the lambda event trigger

        bucket = event['Records'][0]['s3']['bucket']['name']

        key = event['Records'][0]['s3']['object']['key']

        print('Bucket ->', bucket)

        print('Key ->', key)

        print("Hello")

        # Use boto3 to get the JSON file from S3

        response = s3\_client.get\_object(Bucket=bucket, Key=key)

        file\_content = response["Body"].read().decode("utf-8")

        # Read the content into a pandas DataFrame

        data = pd.read\_json(file\_content)

        # Filter records where status is "delivered"

        filtered\_data = data[data['status'] == 'delivered']

        # Convert the filtered DataFrame back to JSON string

        filtered\_json = filtered\_data.to\_json(orient='records')

        # Upload the filtered JSON to the target S3 bucket

        target\_bucket = 'doordash-target-zn-sai'

        target\_key = f'{key.split("/")[-1].split(".")[0]}\_filtered.json'

        s3\_client.put\_object(Body=filtered\_json, Bucket=target\_bucket, Key=target\_key)

        # Publish success message to SNS topic

        sns\_client.publish(

            TopicArn=sns\_topic\_arn,

            Message='Filtered JSON file successfully uploaded to S3.'

        )

        return {

            'statusCode': 200,

            'body': json.dumps('Success')

        }

    except Exception as e:

        # Publish failure message to SNS topic

        sns\_client.publish(

            TopicArn=sns\_topic\_arn,

            Message=f'Error processing file: {str(e)}'

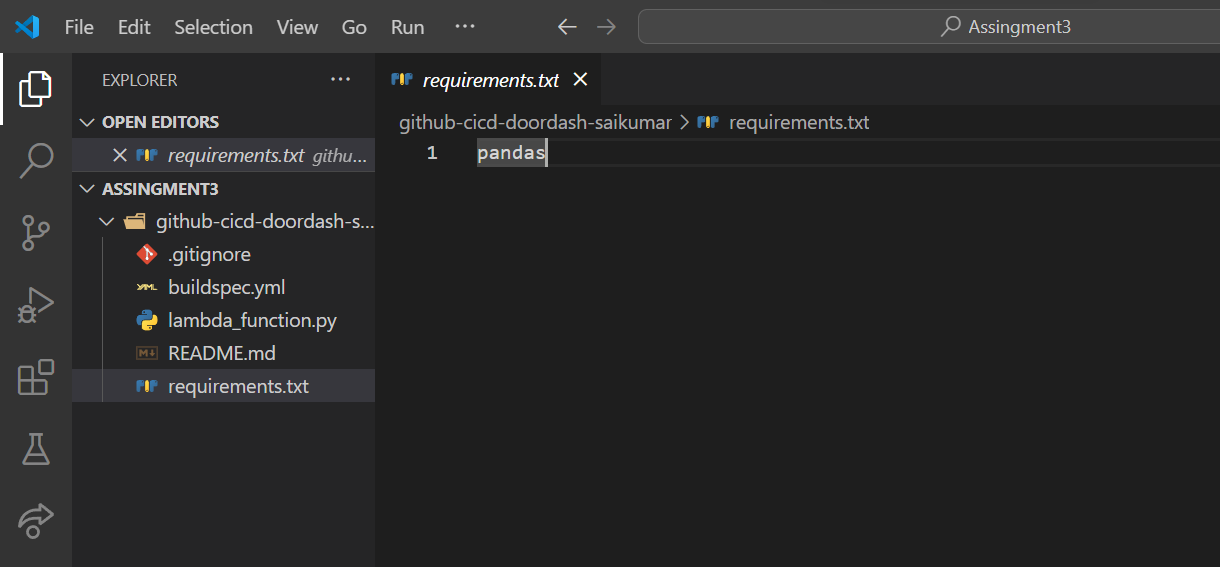
        )

        return {

            'statusCode': 500,

            'body': json.dumps('Error')

        }

Next we will add required libraries or modules needed in requirements.txt file   


Next we will mention commands that are required by codebuild to run automatically , i.e we will mention all the commands in buildspec.yml file below it is

version: 0.2

phases:

  install:

    runtime-versions:

      python: 3.11

    commands:

      - echo "Installing dependencies..."

      - pip install -r requirements.txt -t lib

  build:

    commands:

      - echo "Zipping deployment package..."

      - cd lib

      - zip -r9 ../deployment\_package.zip .

      - cd ..

      - zip -g deployment\_package.zip lambda\_function.py

  post\_build:

    commands:

      - echo "Uploading to S3..."

      - aws s3 cp deployment\_package.zip s3://githublambdadoordashsaikumar-zip/

      - echo "Updating Lambda function via S3..."

      - aws lambda update-function-code --function-name github-to-lambda-cicd-doordash-saikumar --s3-bucket githublambdadoordashsaikumar-zip --s3-key deployment\_package.zip

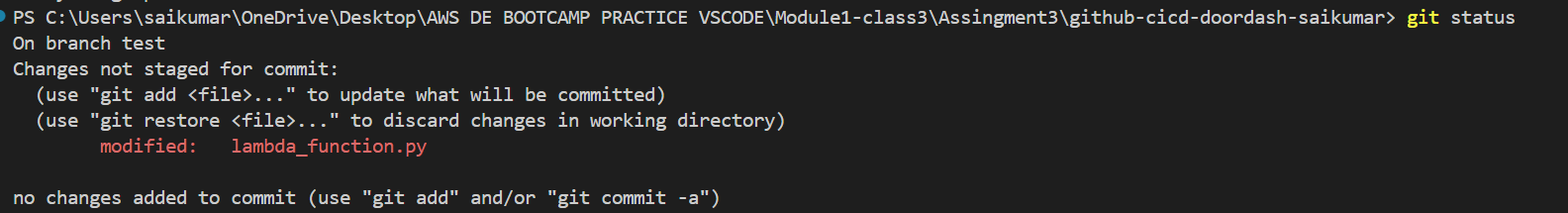
      - echo "Deployment complete!"

6).Next we will

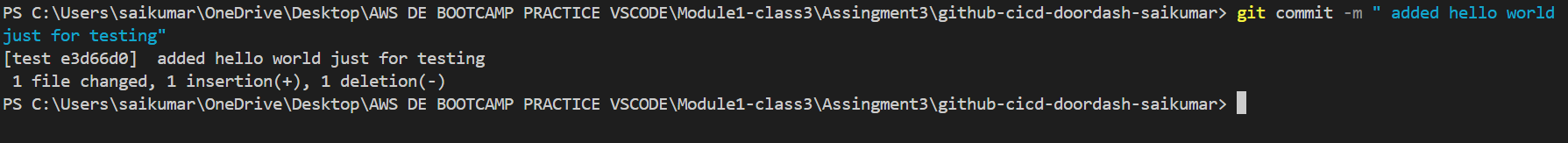
Move all this code to Test branch of GitHub Repo we created.

By Vscode Terminal or By Cmd we can move this code to github by below commands

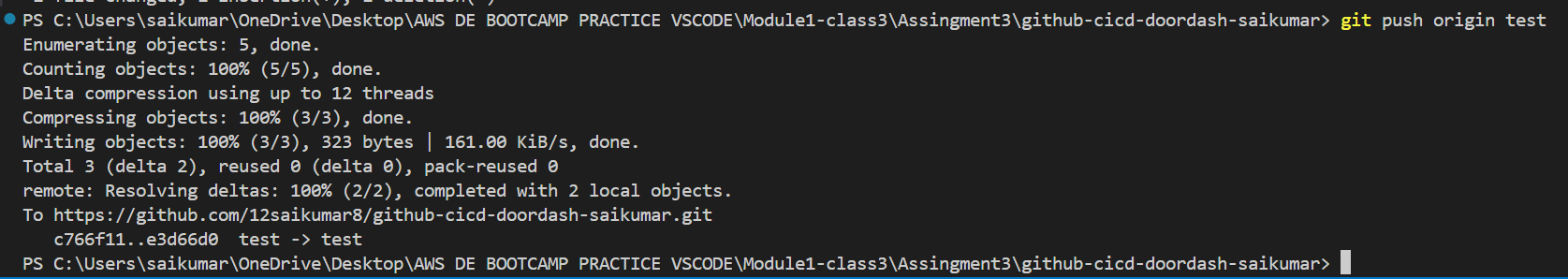


* We can see git status to see unadded files
* 
* Next Git add . to add all the files
* 

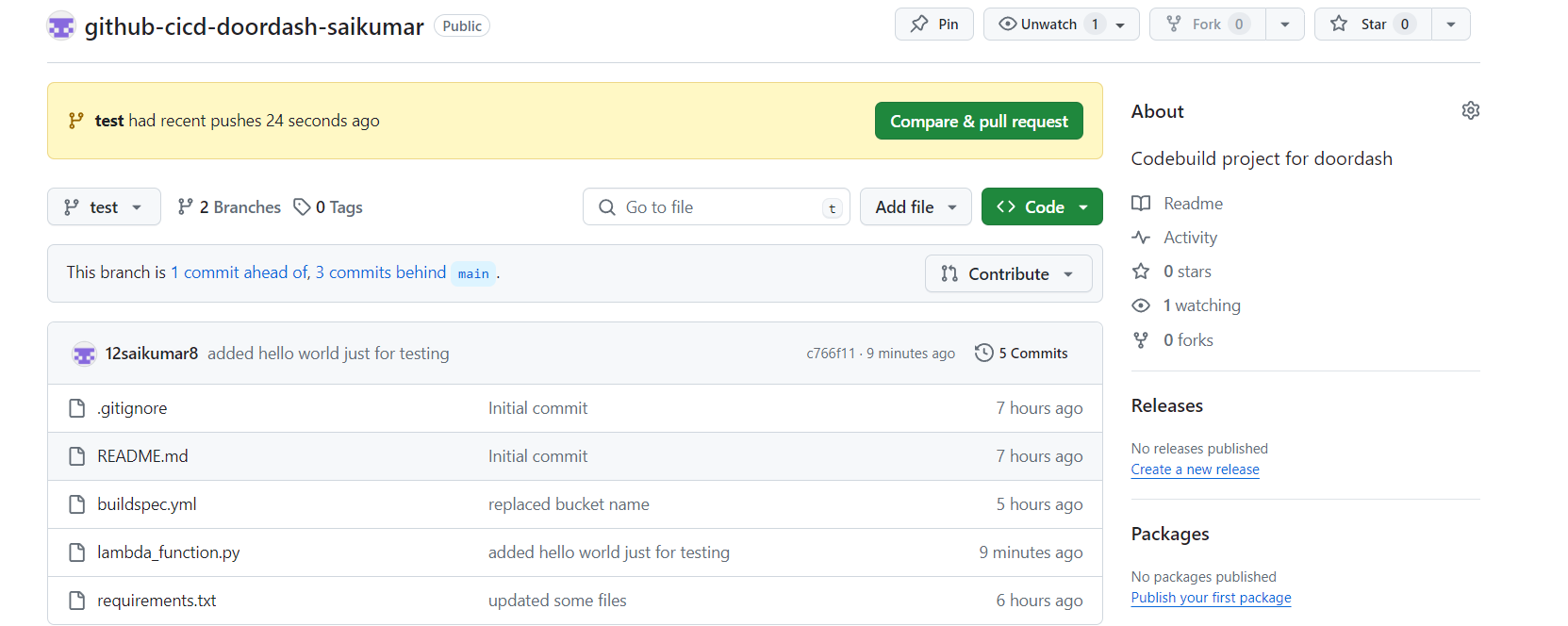
🡪Next git commit for commiting files



🡪Next Git push origin test to push to github

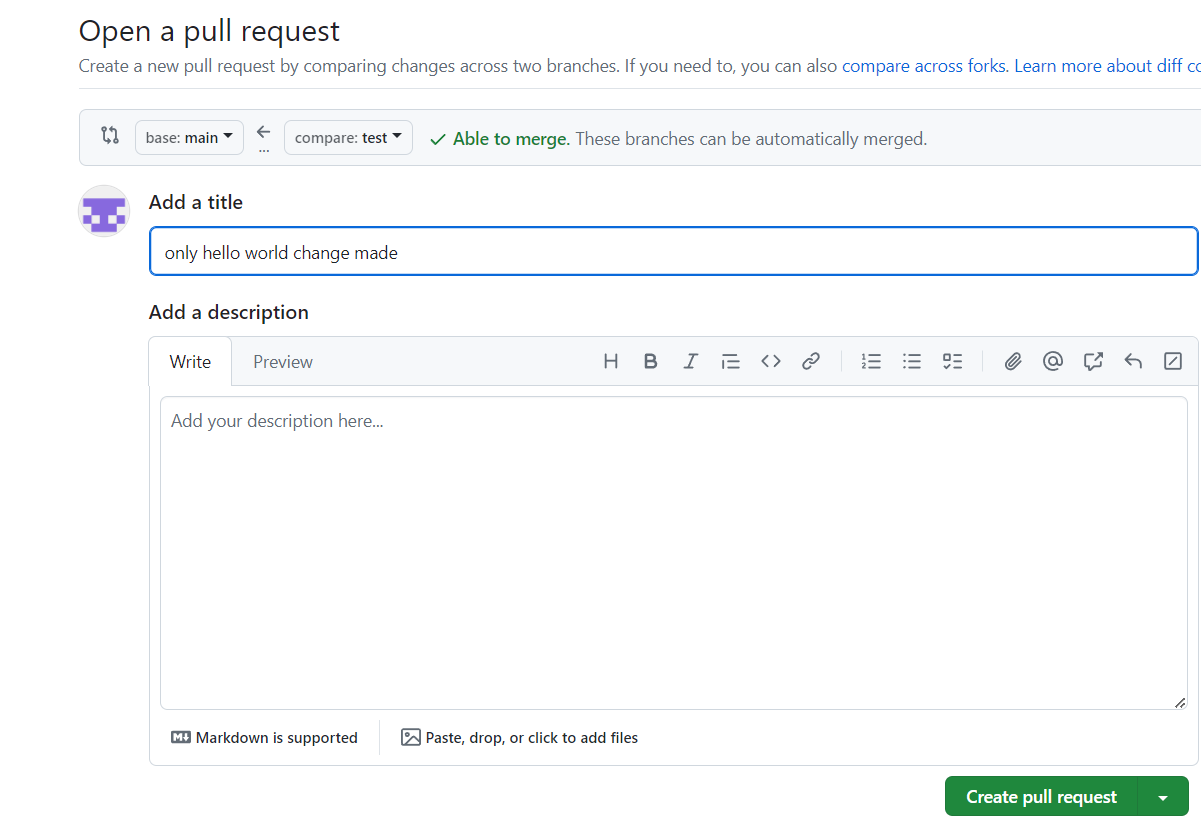


Now go to github test branch and we can see all are available there



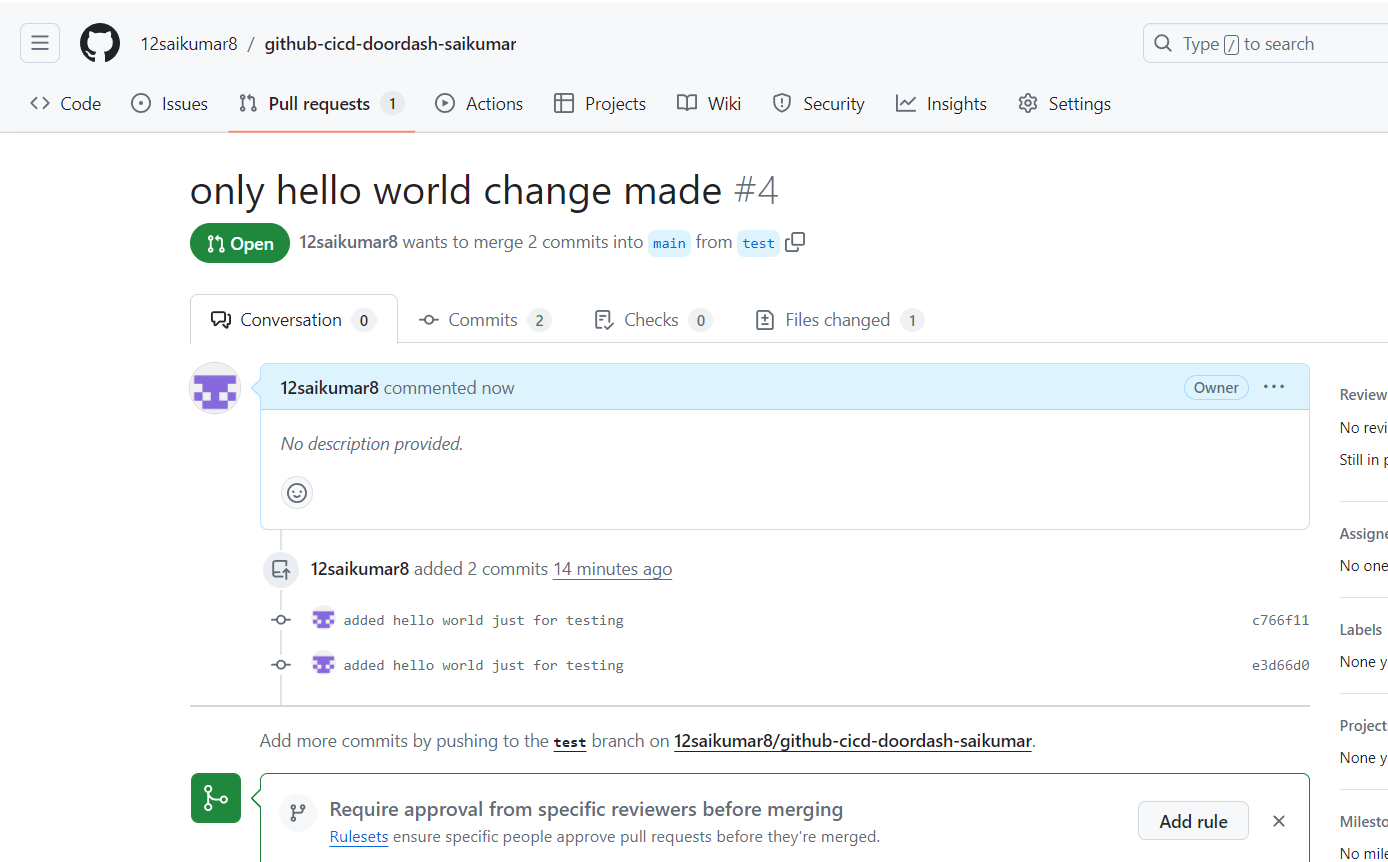
7). Next will Click on Compare &Pull request Above for creating Pull request and will Merge this code to main Branch

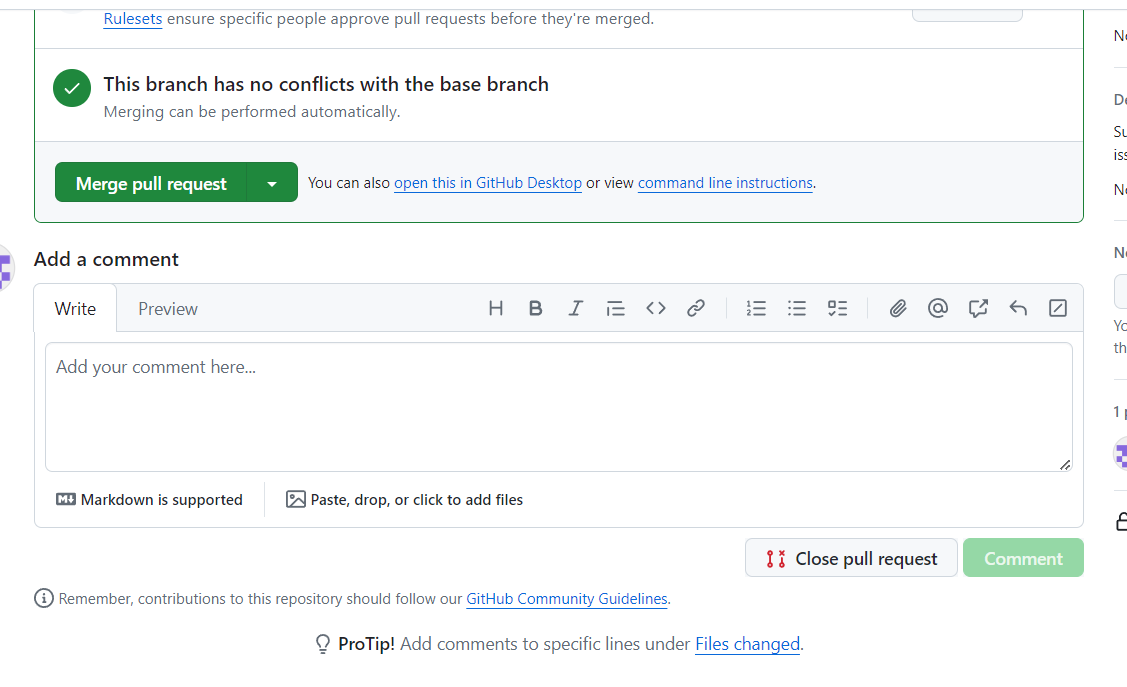
Next below screenshot opens



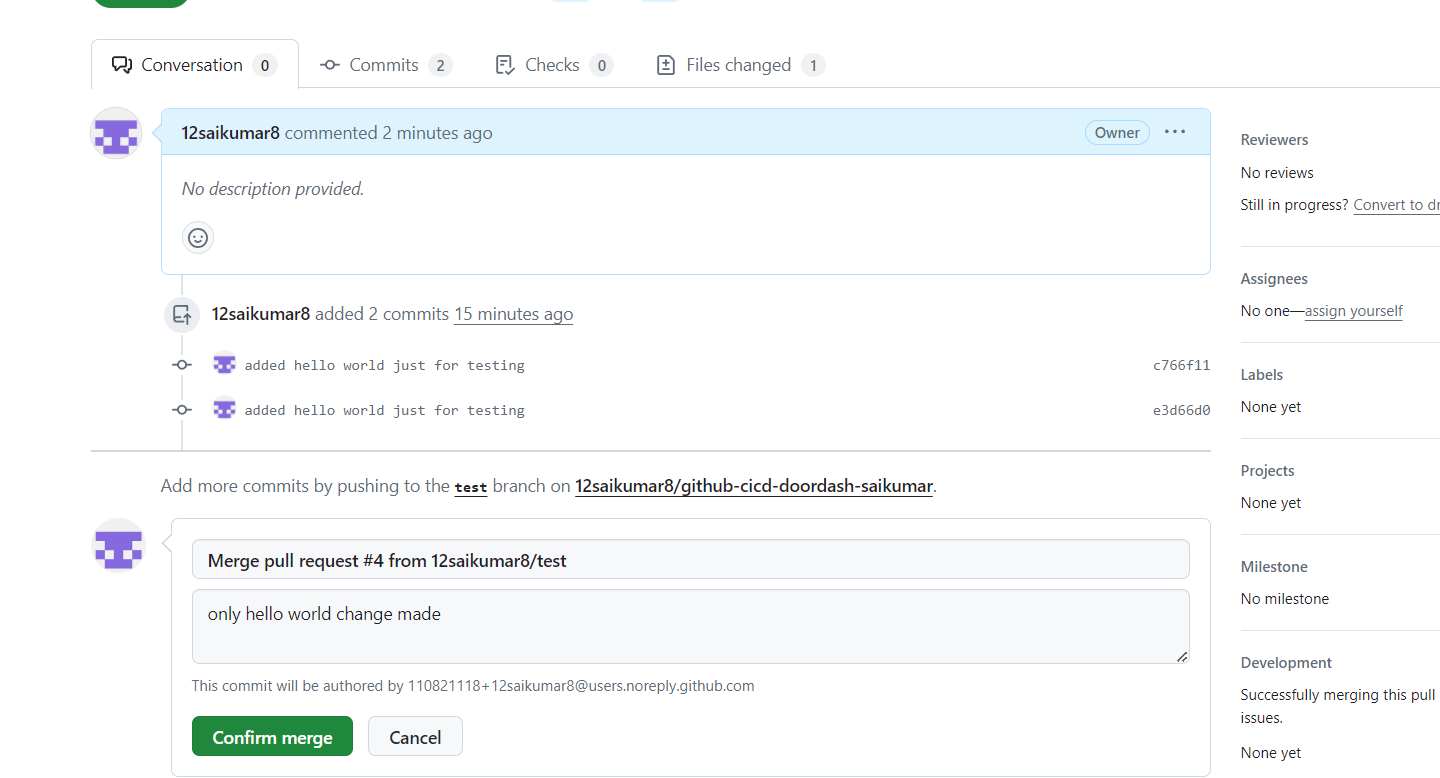
Click on create Pull request

Next below click on merge pull request and

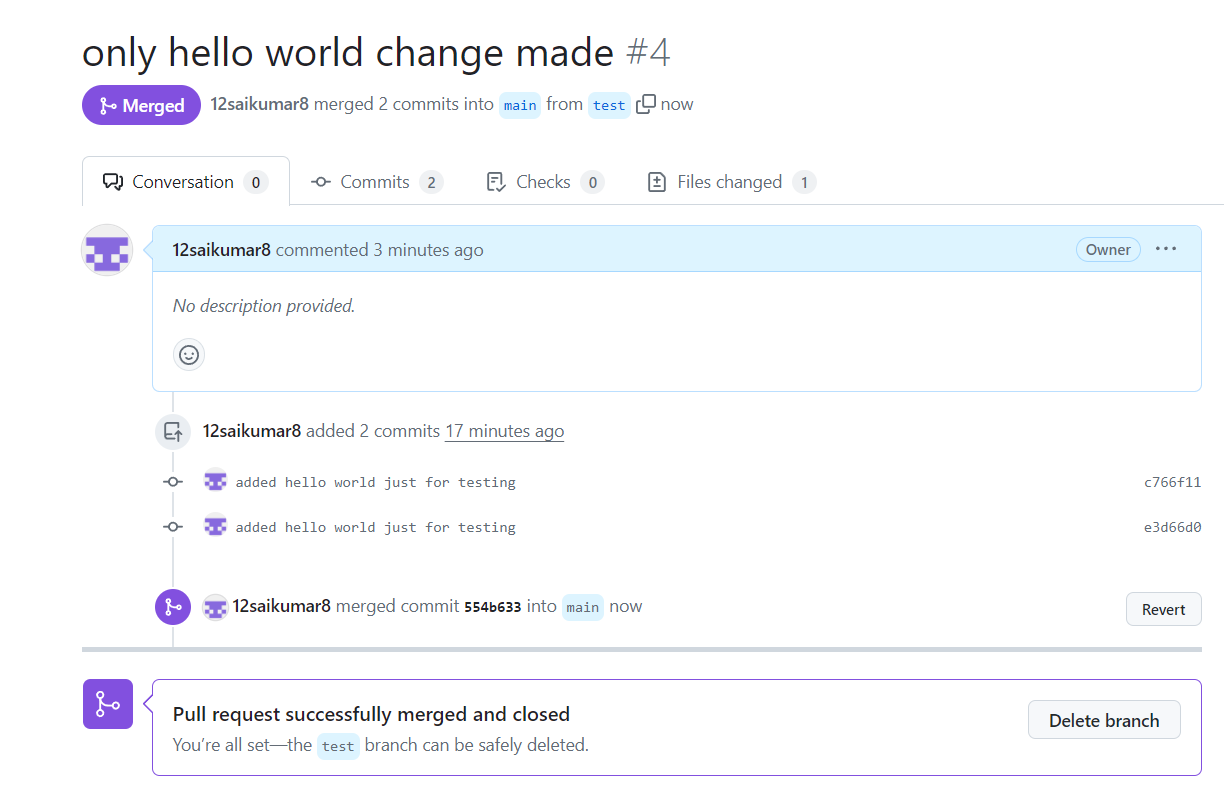




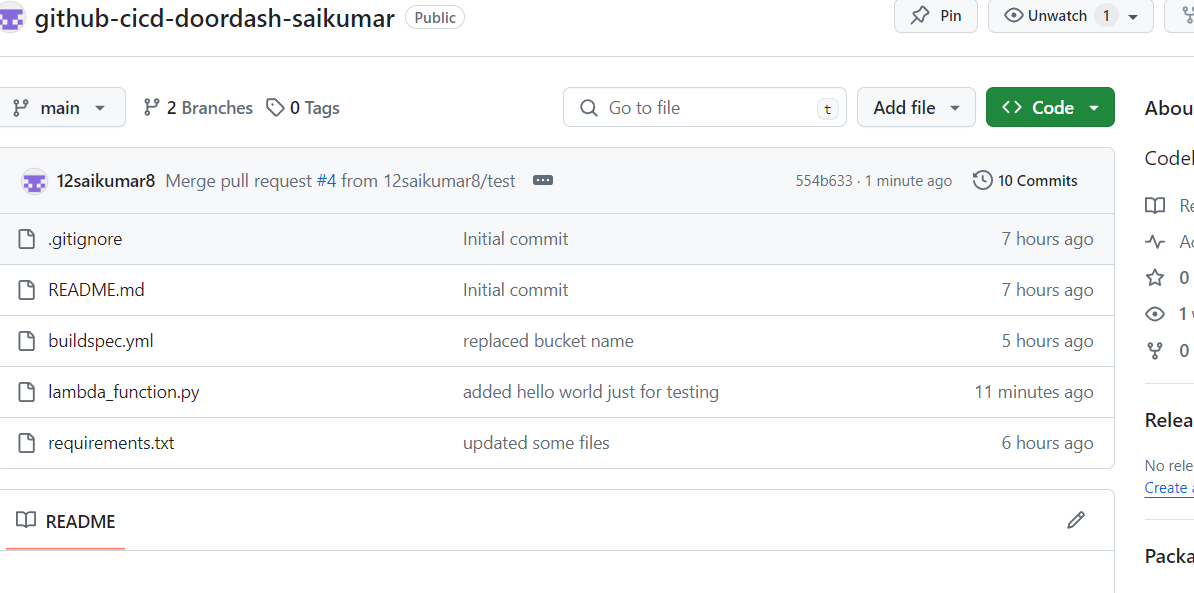
And click on confirm



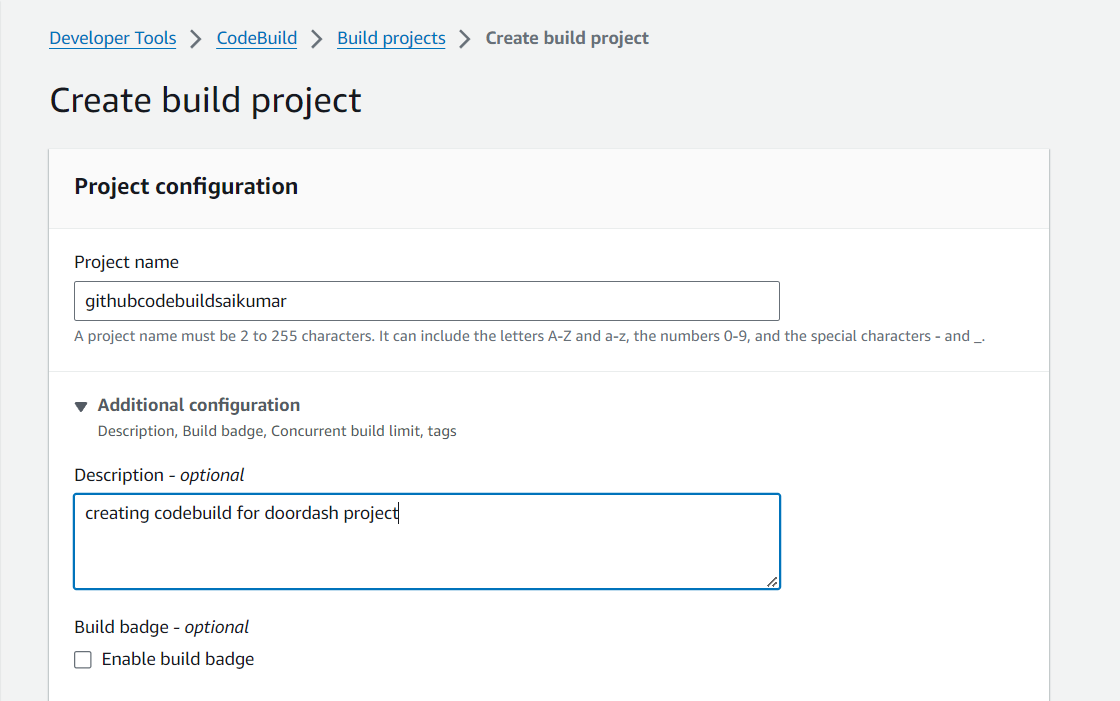
* Next we can see it’s merged



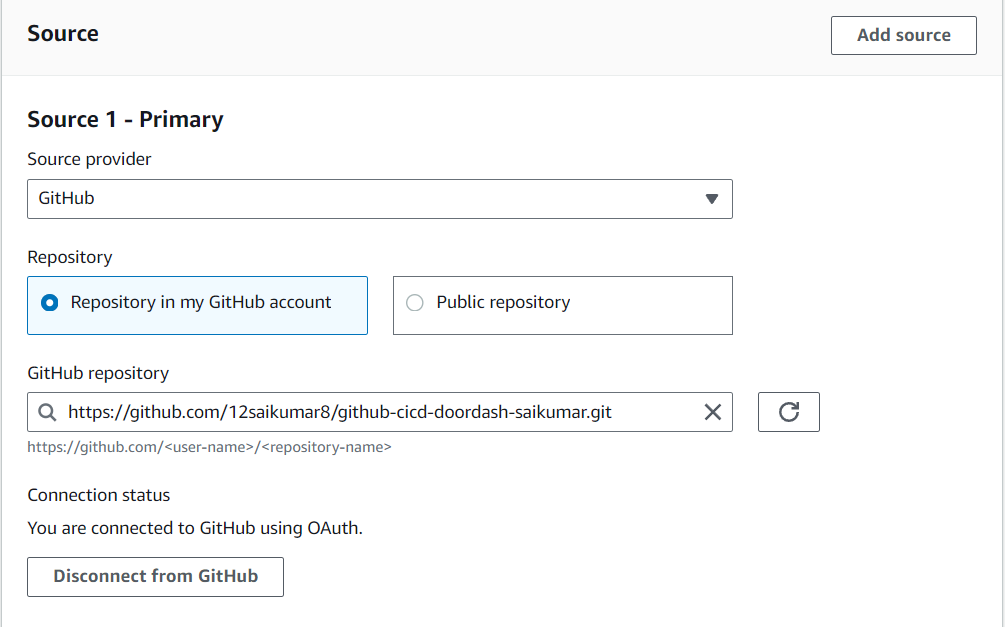
8). WE can check on main branch has received all files



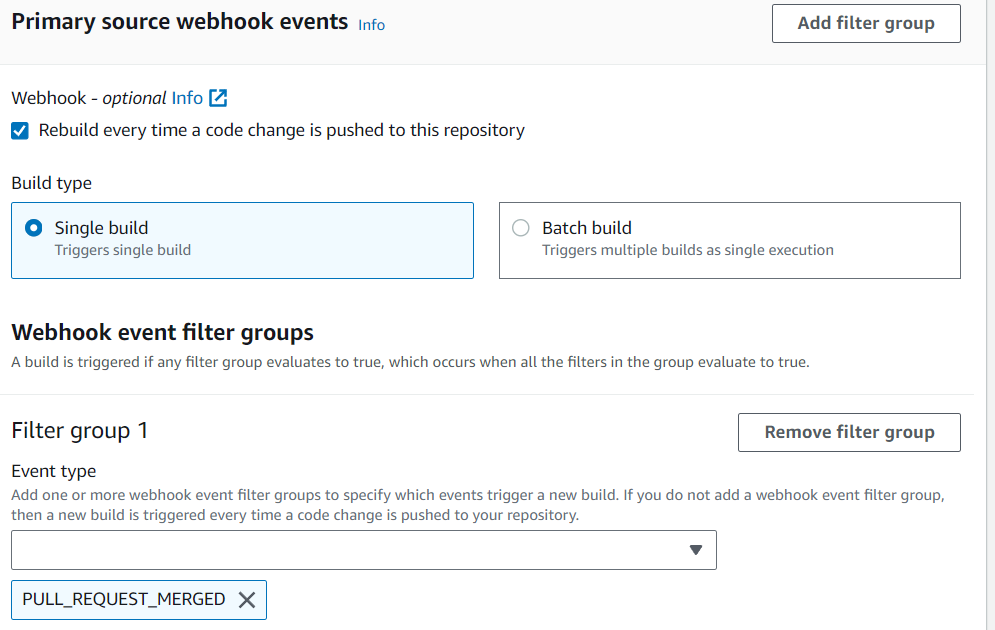
9).Next will create code build when creating we will add our git repo to this



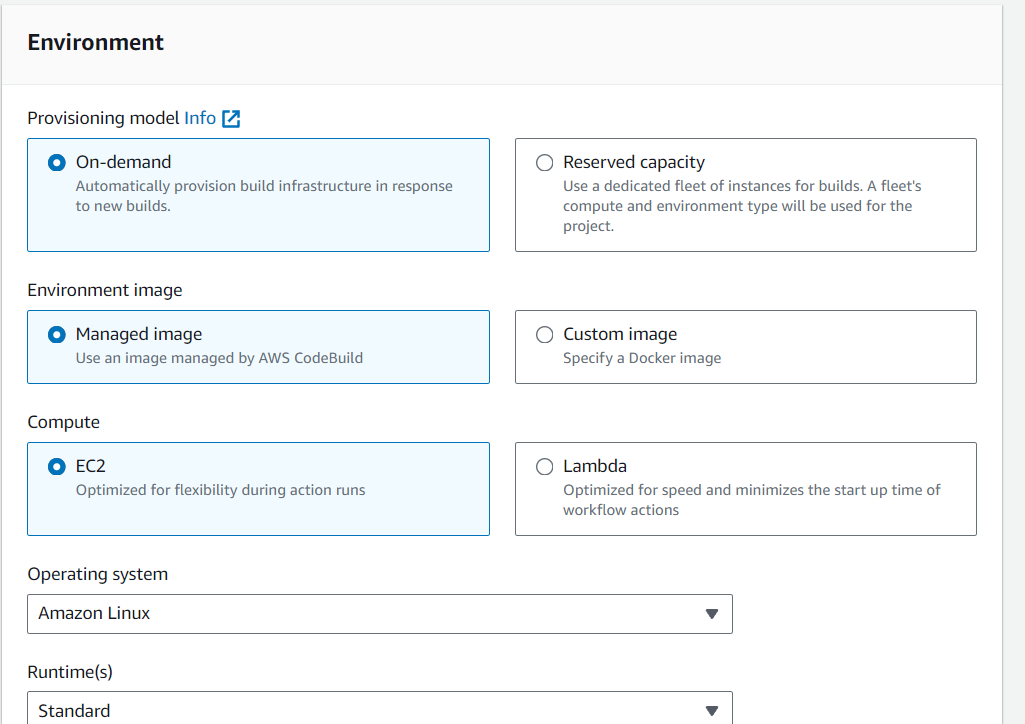
Next source will add

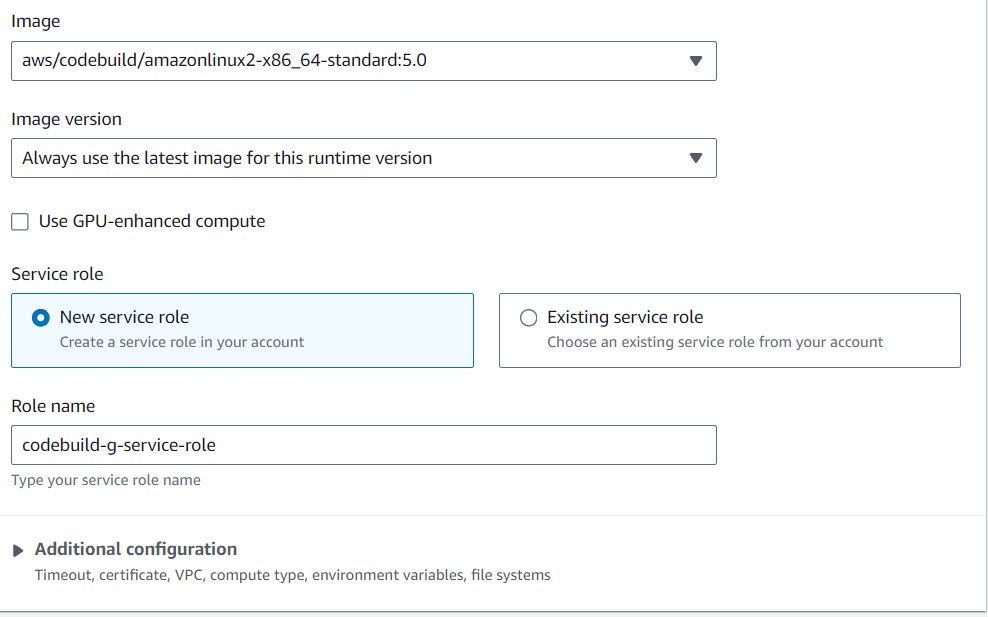


Next will tick webhook and in filter group pull\_request\_merged

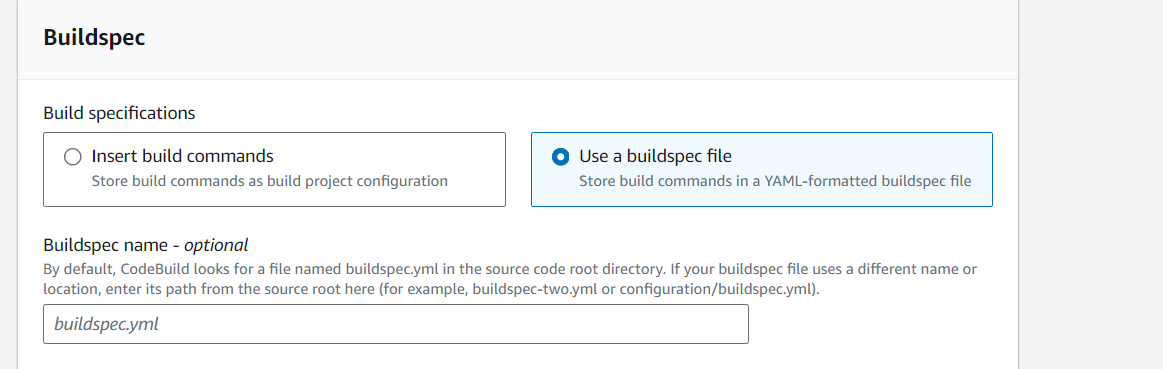


Next these envirorment depending on configuration

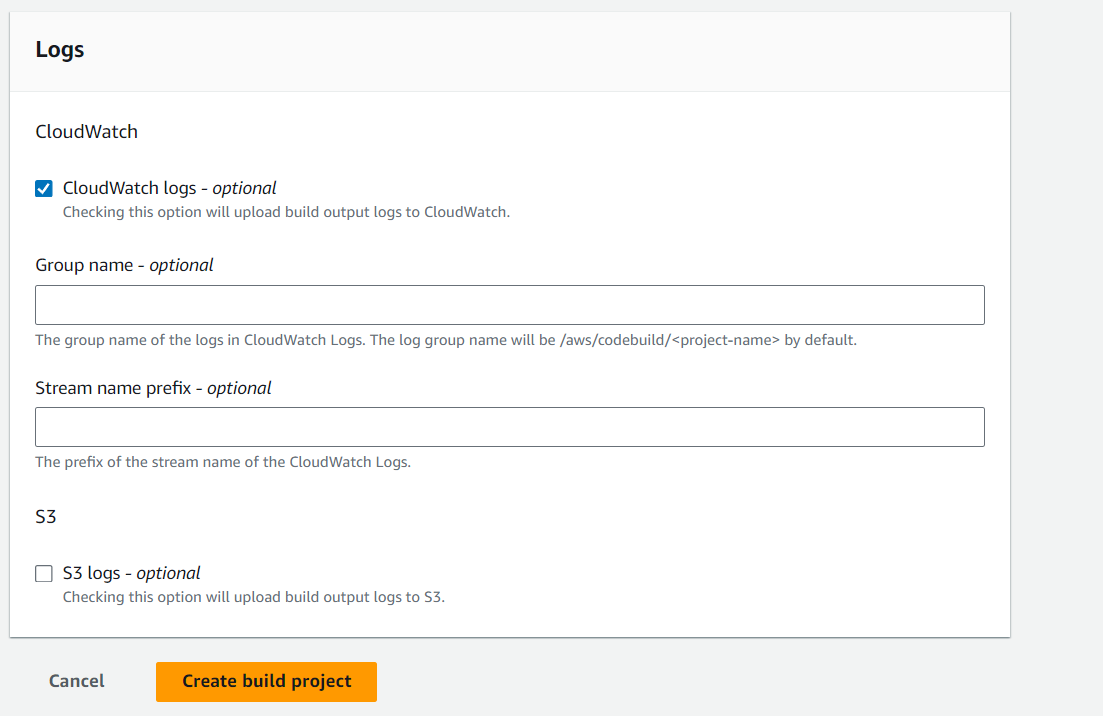




Next we will choose buildspec

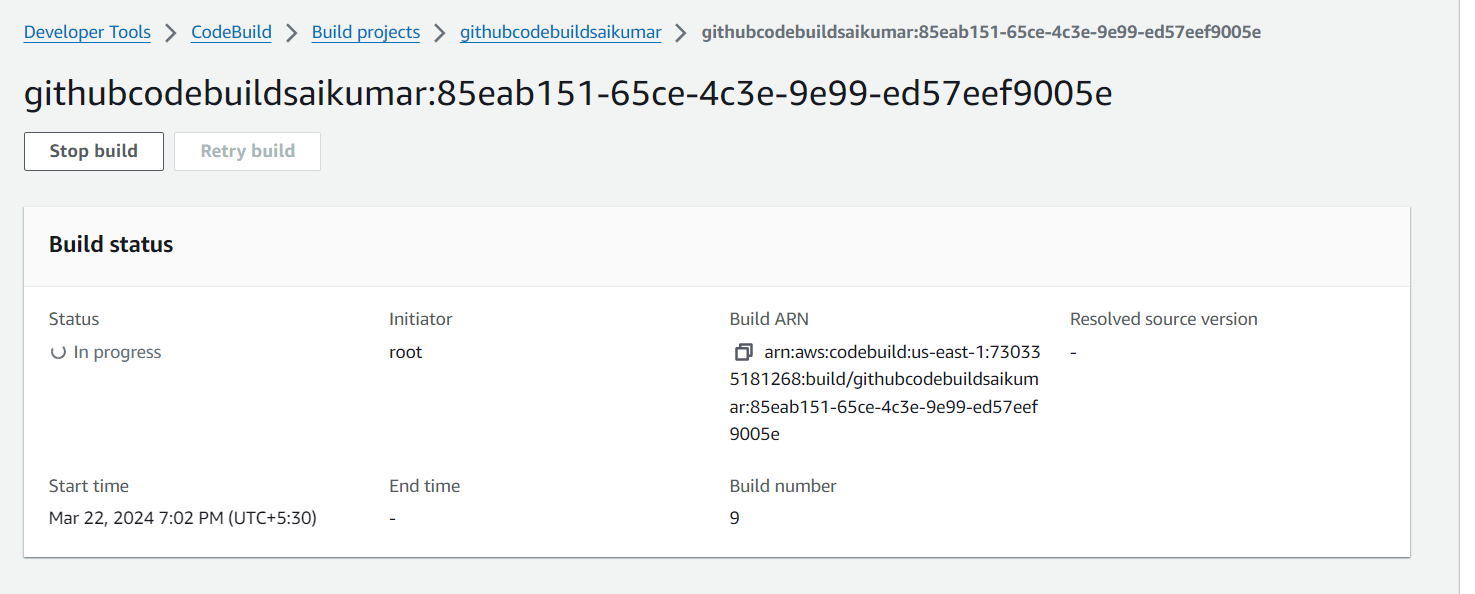


Finally click on create

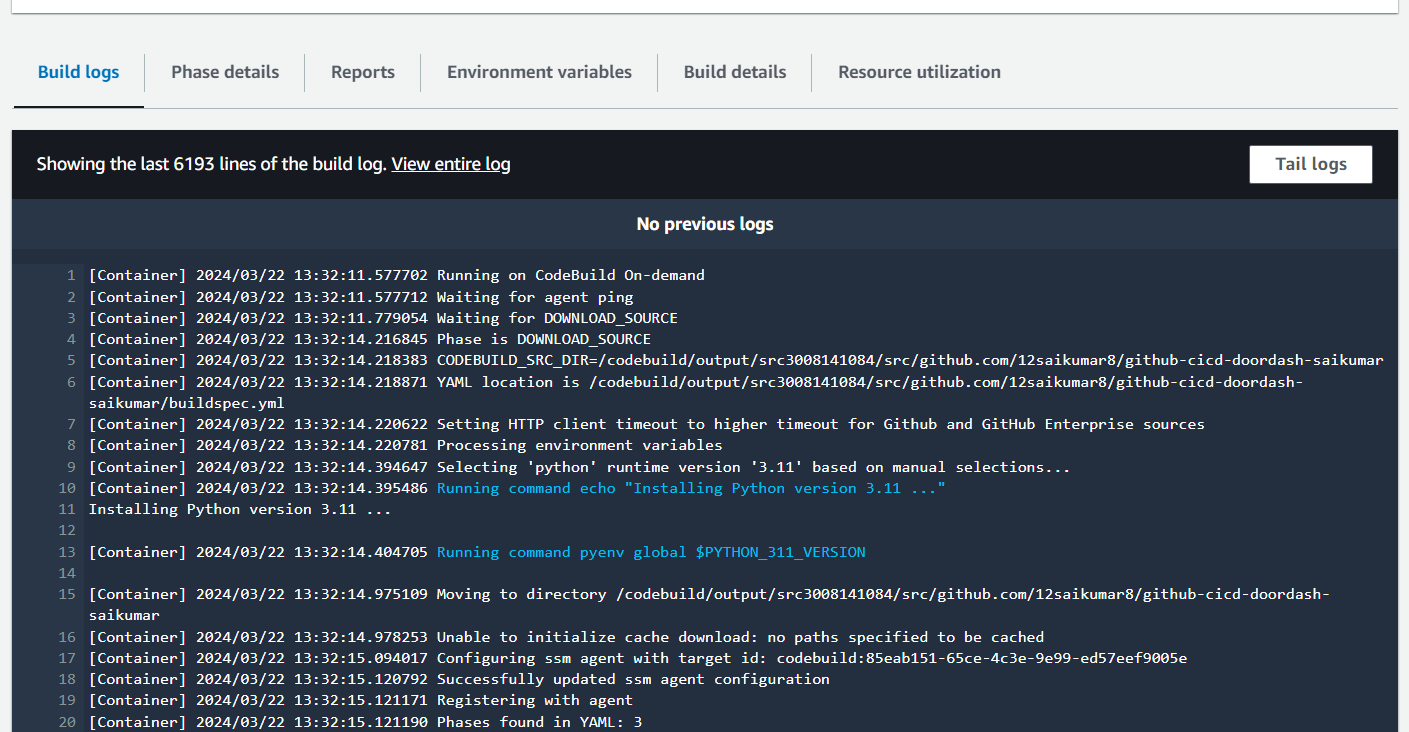


It is created

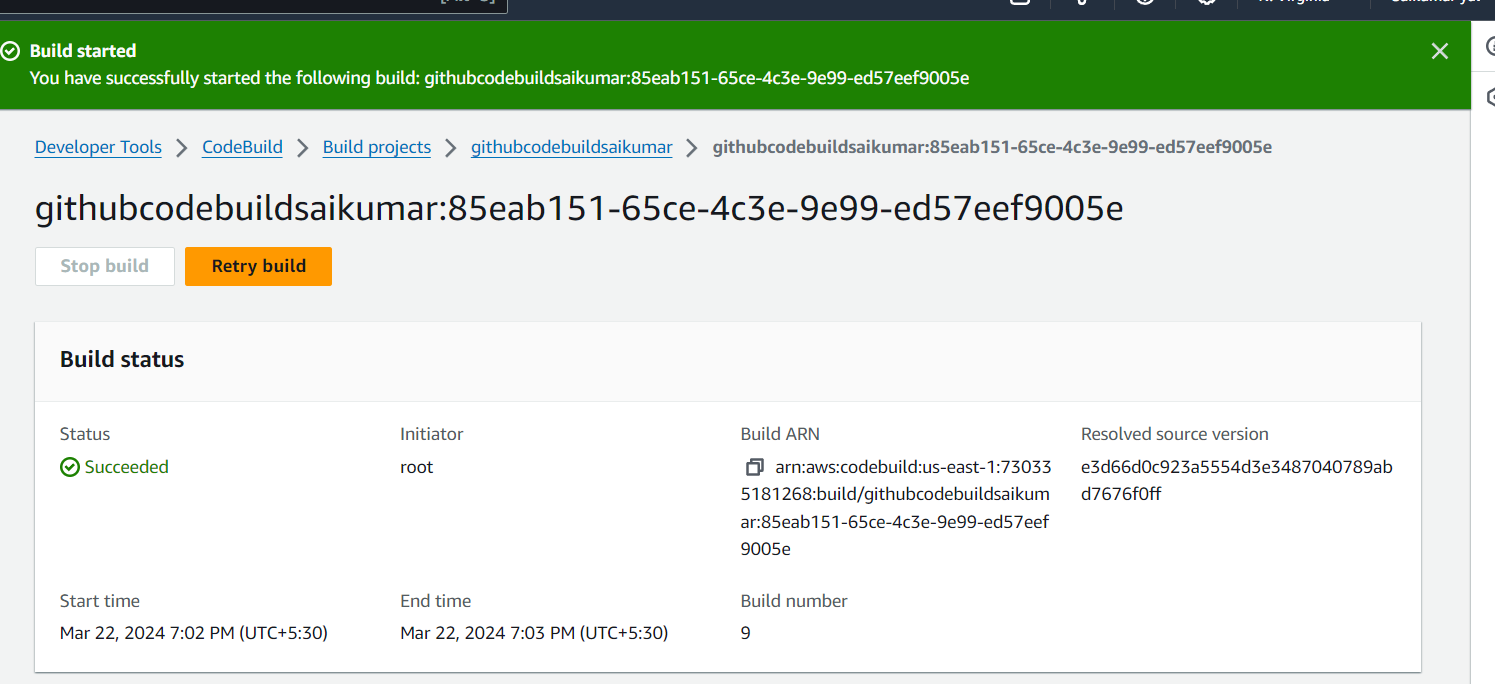
🡪10). Next we can see it is running



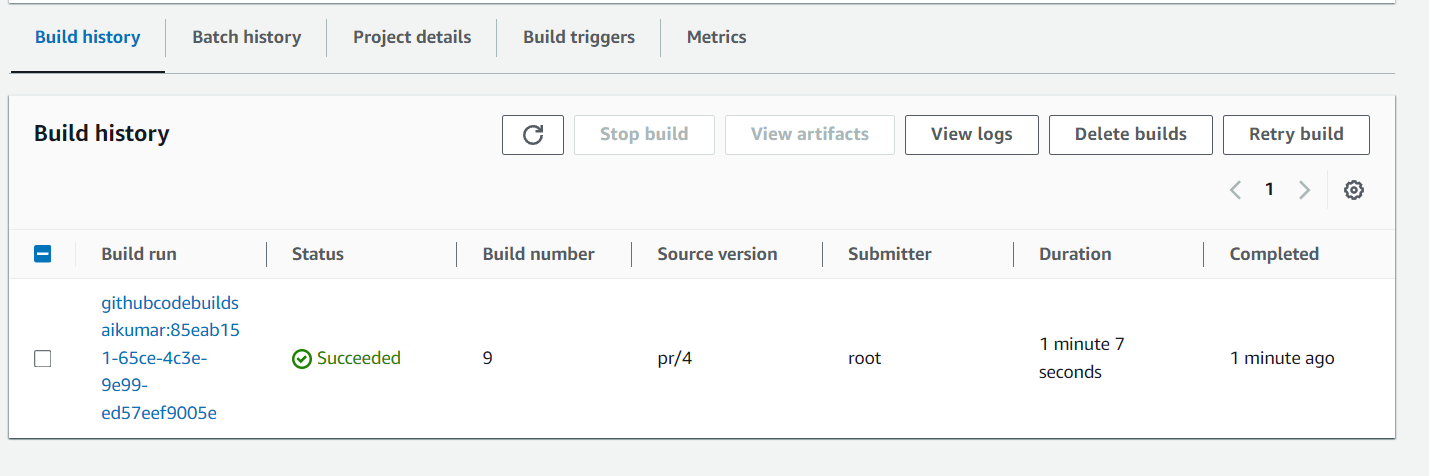
And below we can see all build logs



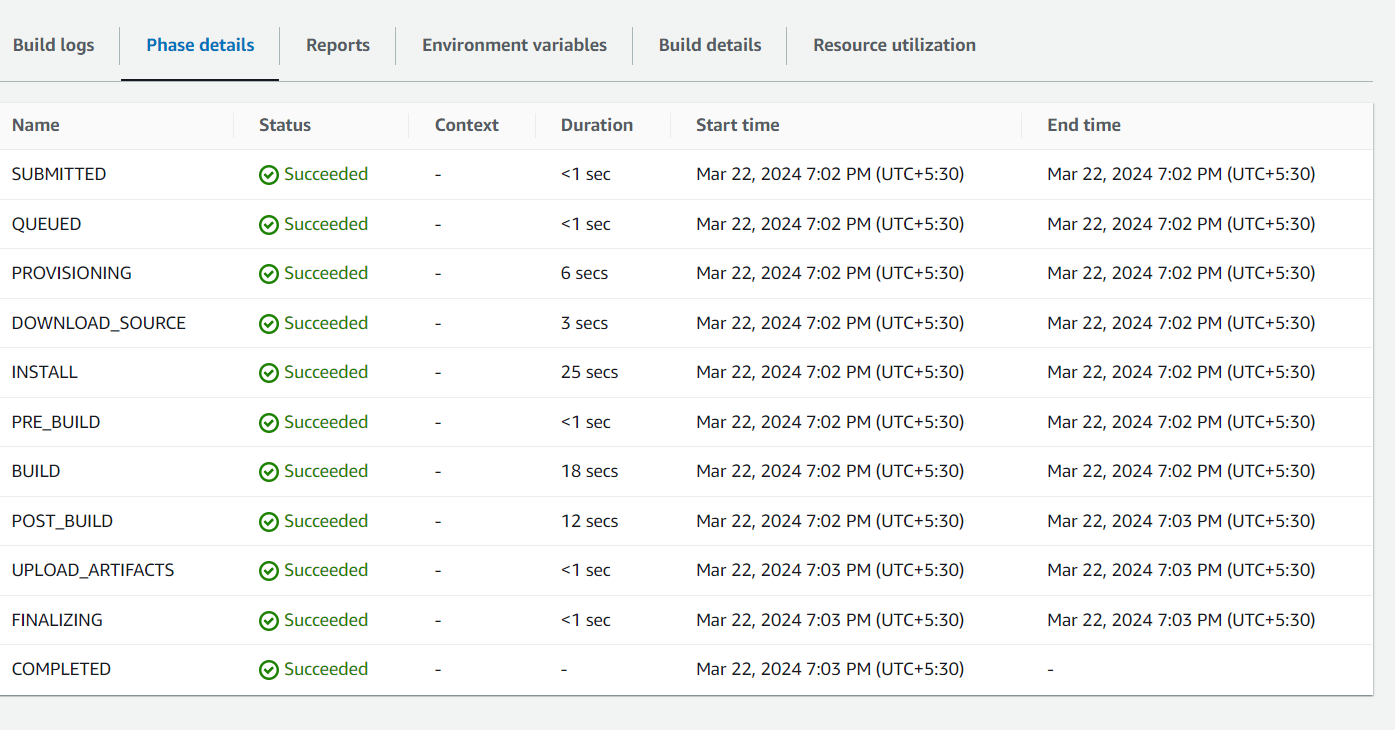
* We can see it’s succeded



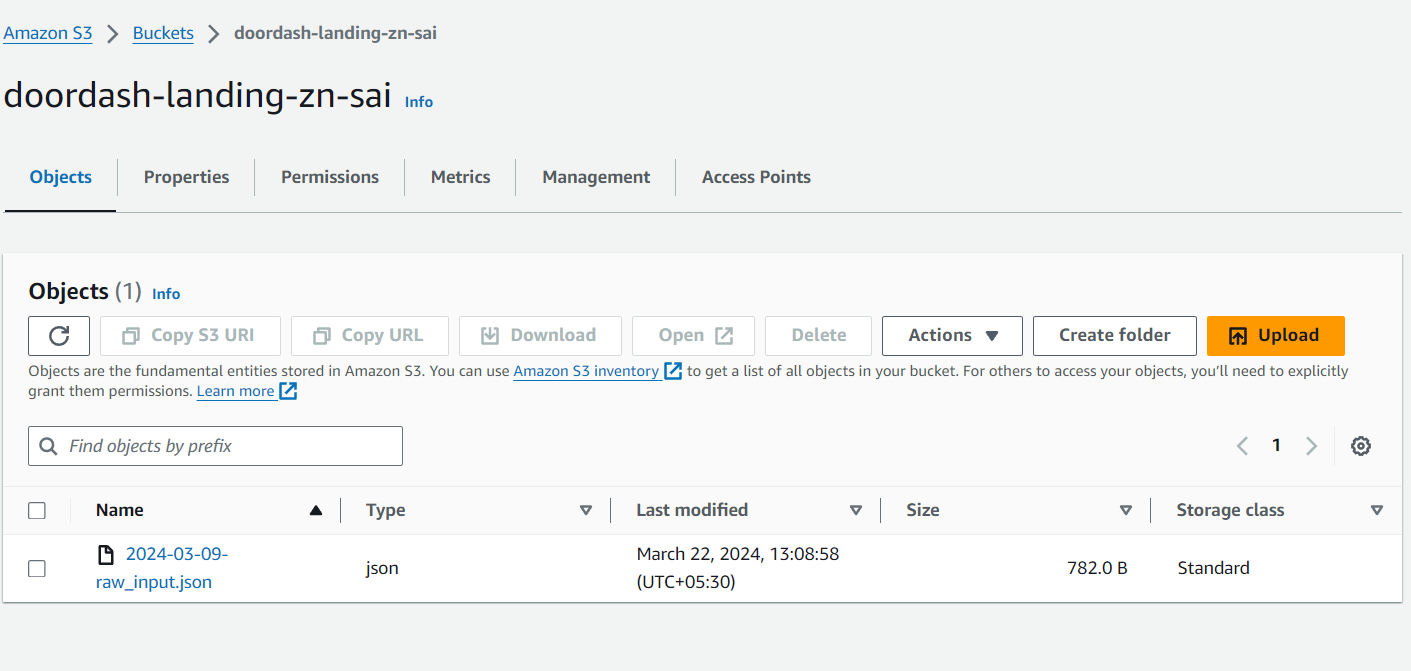
Build History



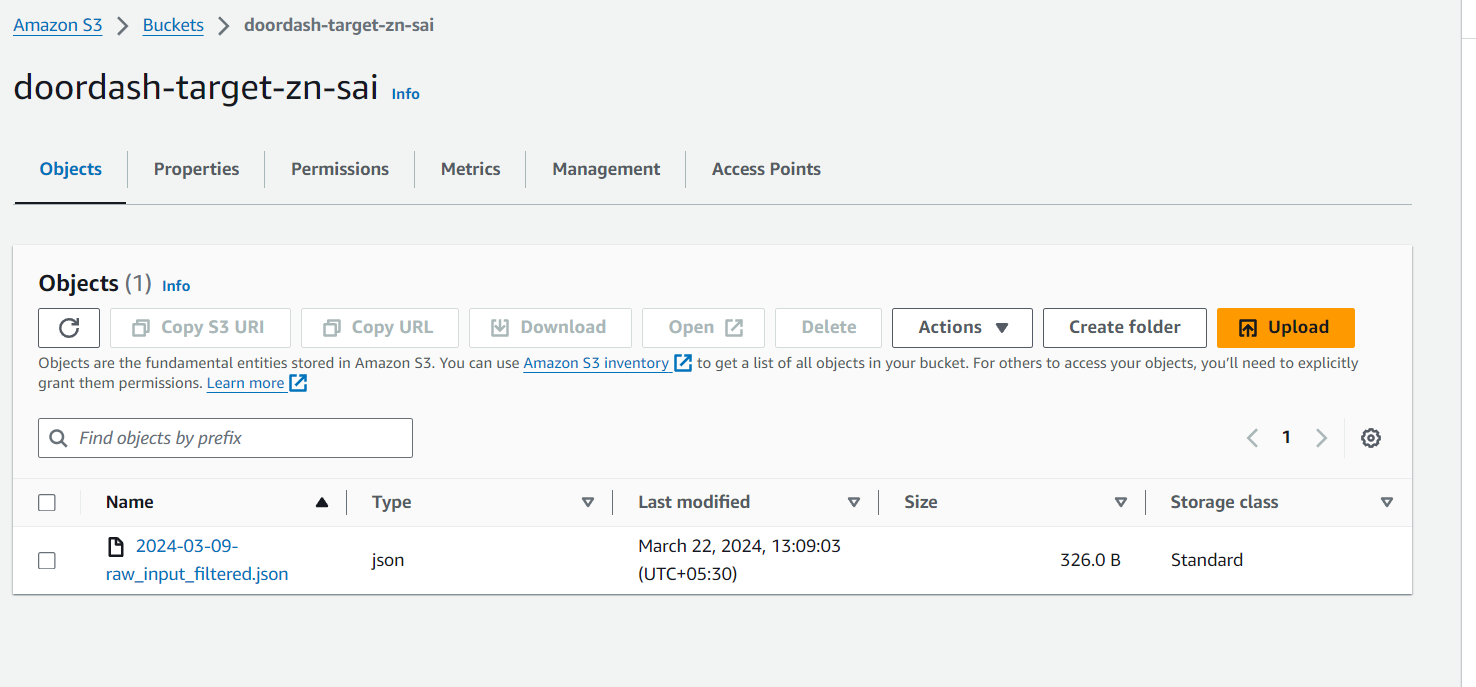
We can see phase details for details



11).So now we will upload a raw file( [2024-03-09-raw\_input.json](https://s3.console.aws.amazon.com/s3/object/doordash-landing-zn-sai?region=us-east-1&bucketType=general&prefix=2024-03-09-raw_input.json)) file to s3bucket that added as trigger below



12).And we can see we received filtered or result file to the target bucket(doordash-target-zn-sai)



The result/Filtered file received is ([2024-03-09-raw\_input\_filtered.json](https://s3.console.aws.amazon.com/s3/object/doordash-target-zn-sai?region=us-east-1&bucketType=general&prefix=2024-03-09-raw_input_filtered.json))

13).And next we can check our mail as we subscribed to SNS Topic so we will receive notification to mail