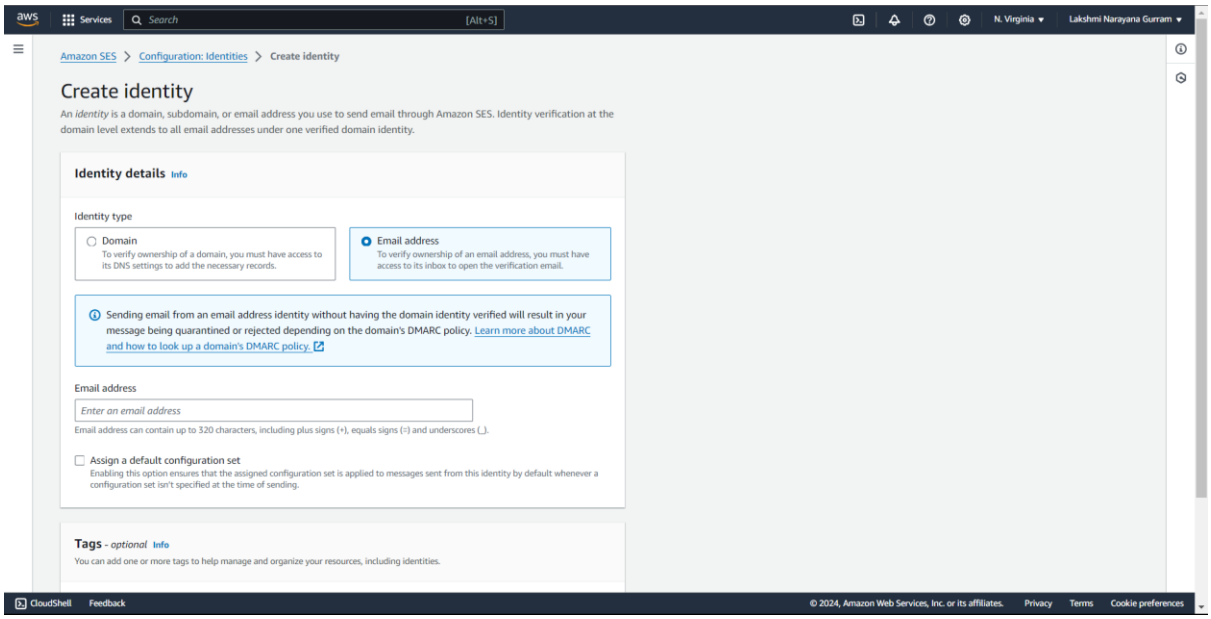
Remainder Application using AWS Serverless

2100031943

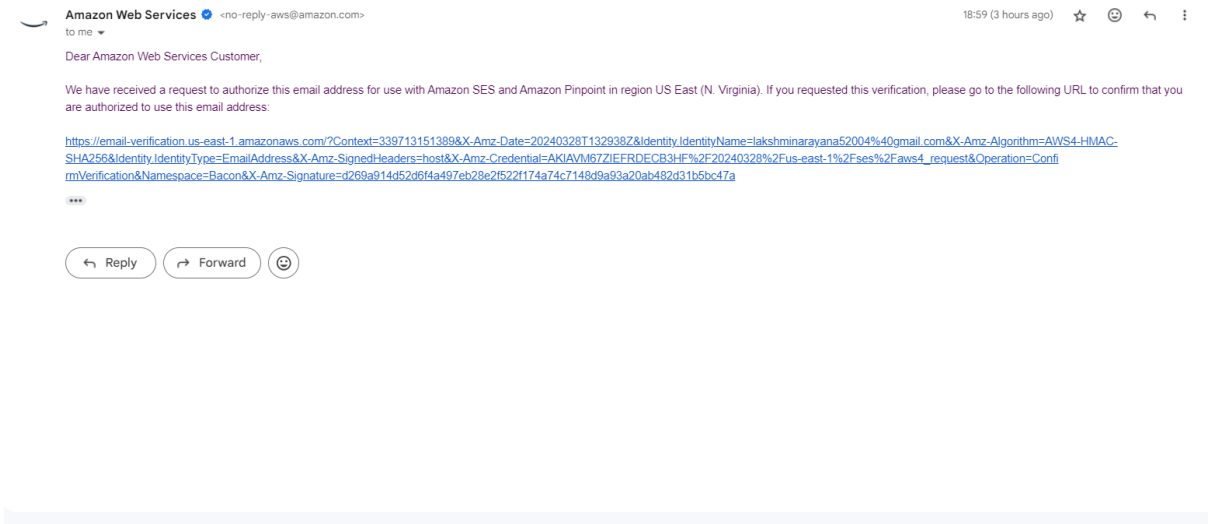
G Lakshmi Narayana

Open SES service and create a identity

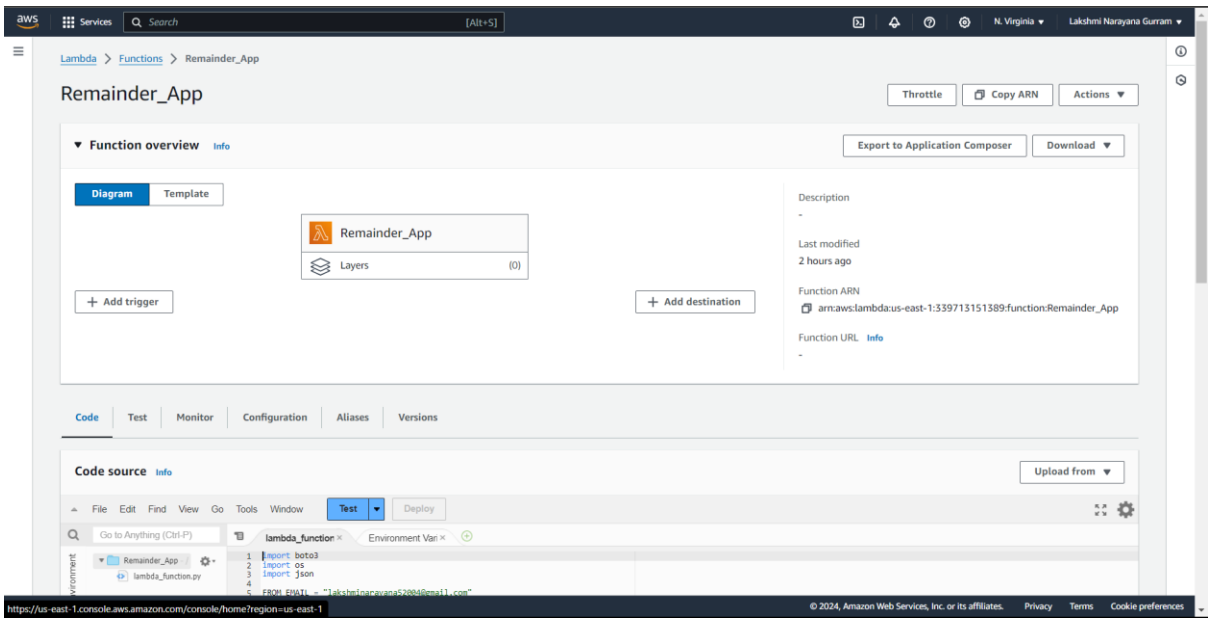
And select email



and verify the mail



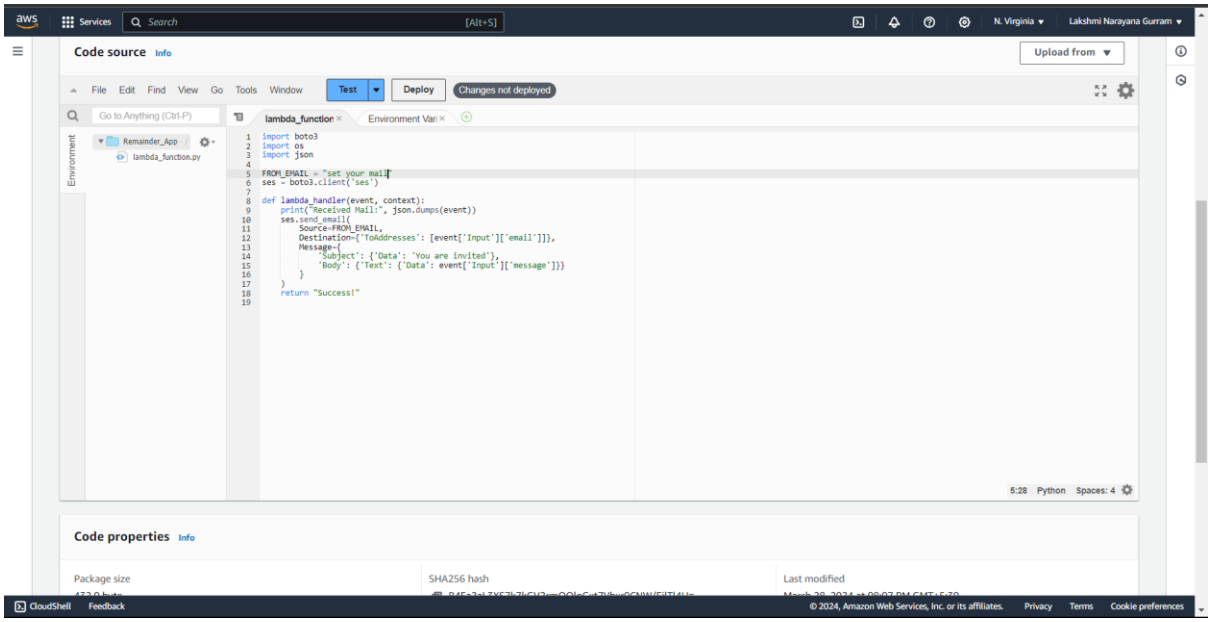
Create a Lambda function
with run time python



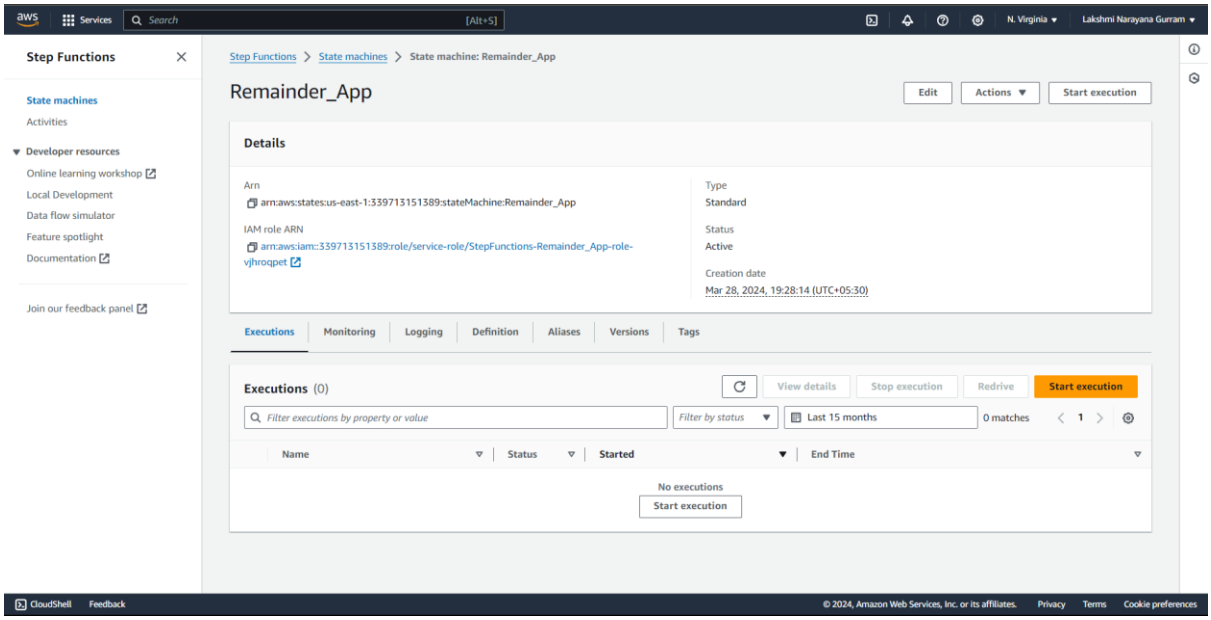
Code:

```
import boto3

import os

import json


FROM_EMAIL = "set your mail"

ses = boto3.client('ses')


def lambda_handler(event, context):

  print("Received Mail:", json.dumps(event))

  ses.send_email(

    Source=FROM_EMAIL,

    Destination={'ToAddresses': [event['Input']['email']]},

    Message={

      'Subject': {'Data': 'You are invited'},

      'Body': {'Text': {'Data': event['Input']['message']}}
```

```
    }
)
return "Success!"
```



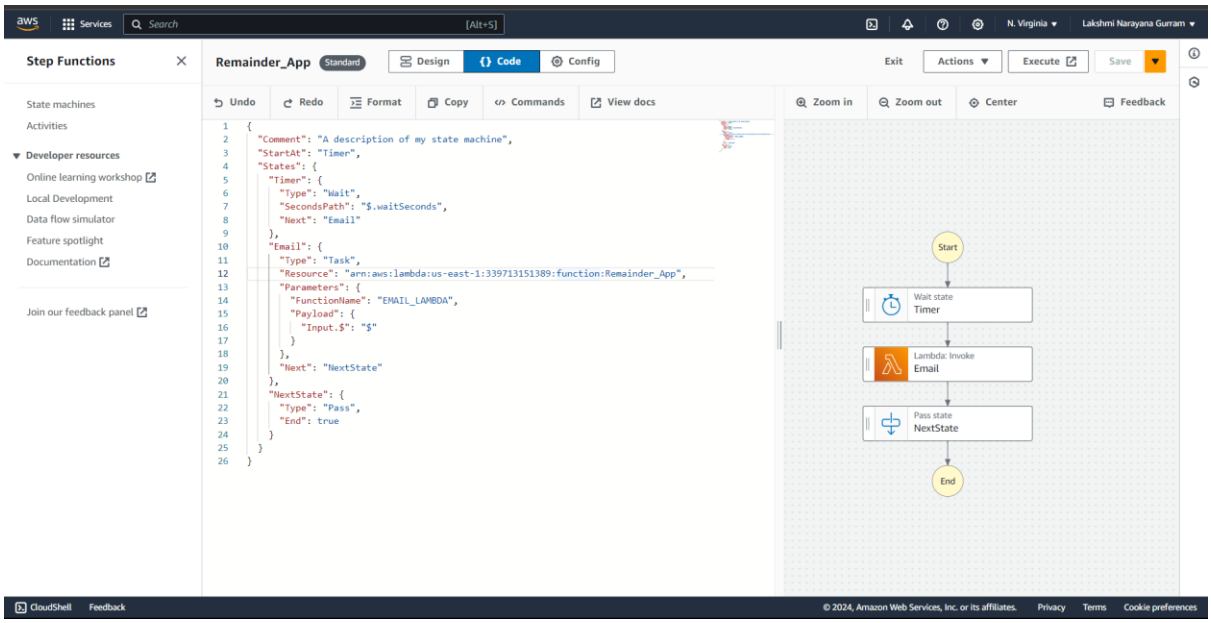
Create a step function

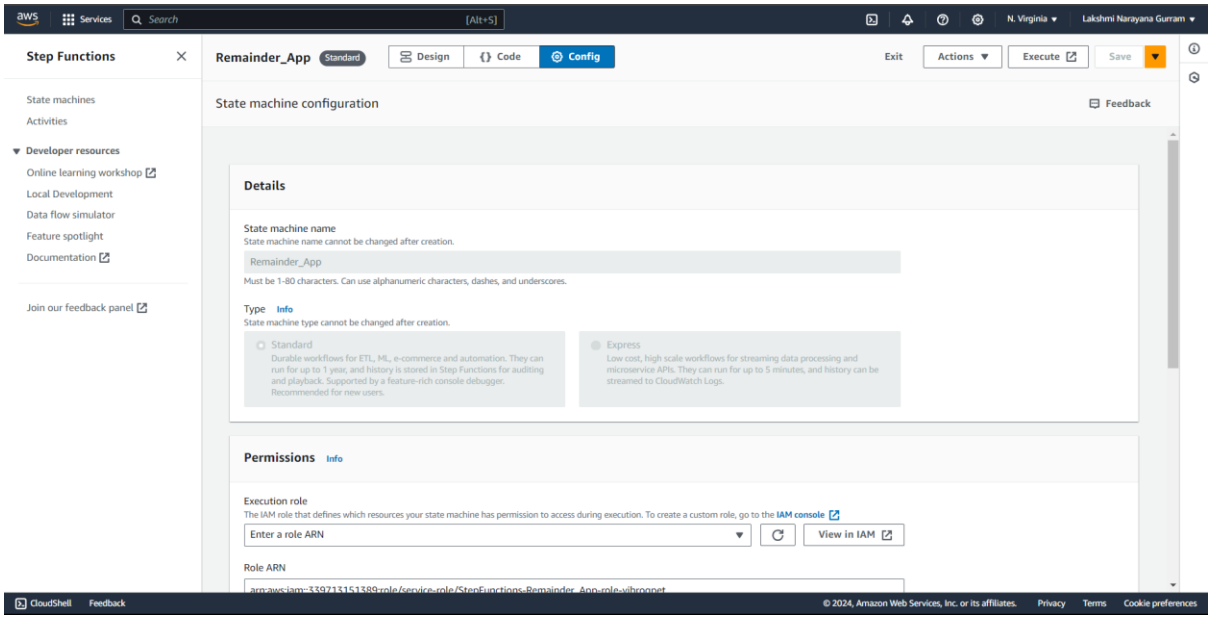


Code:

```
{
  "Comment": "A description of my state machine",
  "StartAt": "Timer",
  "States": {
    "Timer": {
      "Type": "Wait",
```

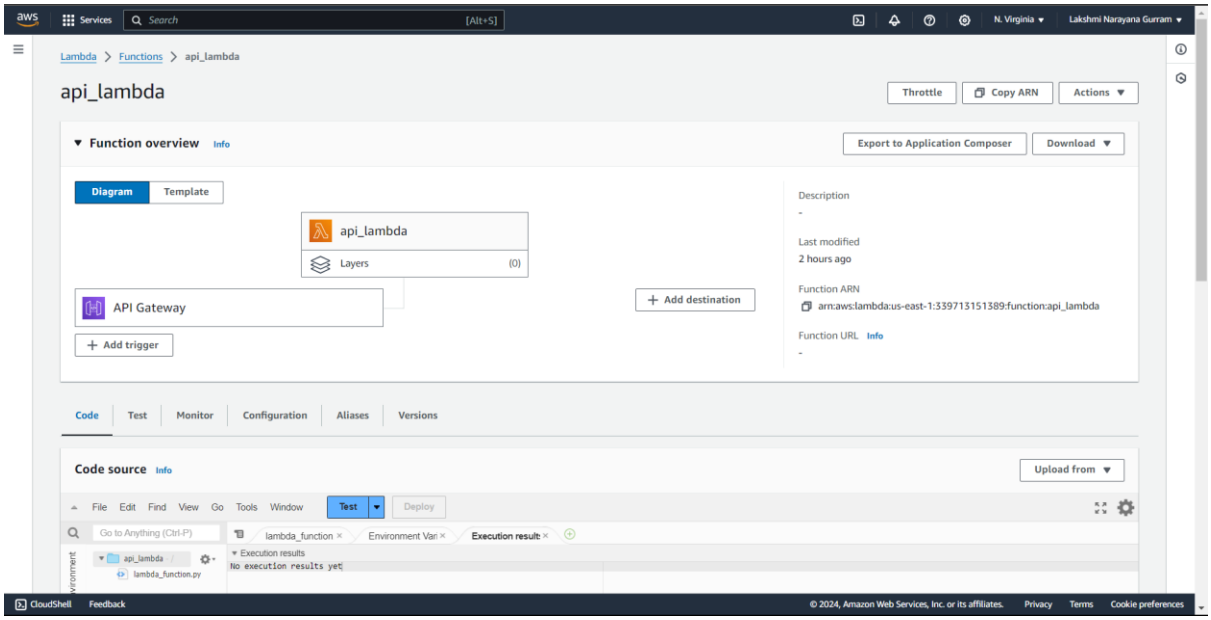
```
      "SecondsPath": "$.waitSeconds",
      "Next": "Email"
    },
    "Email": {
      "Type": "Task",
      "Resource": "lambda function arn",
      "Parameters": {
        "FunctionName": "EMAIL_LAMBDA",
        "Payload": {
          "Input.$": "$"
        }
      },
      "Next": "NextState"
    },
    "NextState": {
      "Type": "Pass",
      "End": true
    }
  }
}
```





create another lambda function

run time python



Code: import boto3

import json

SM_ARN = “stepdfunction_Arn”

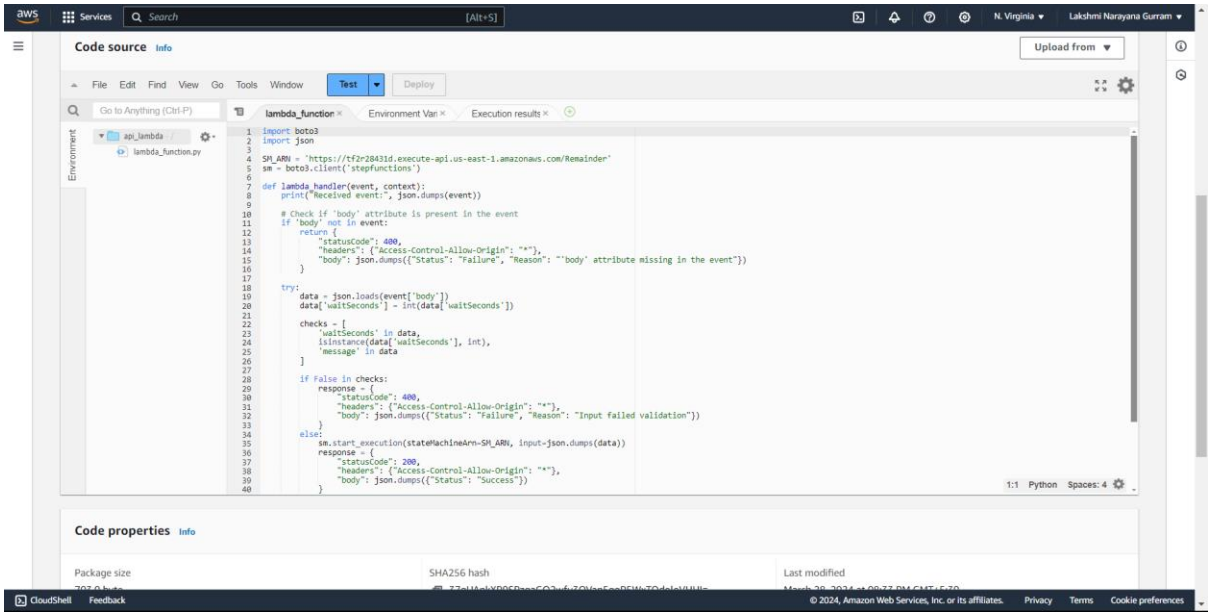
sm = boto3.client('stepfunctions')

def lambda_handler(event, context):

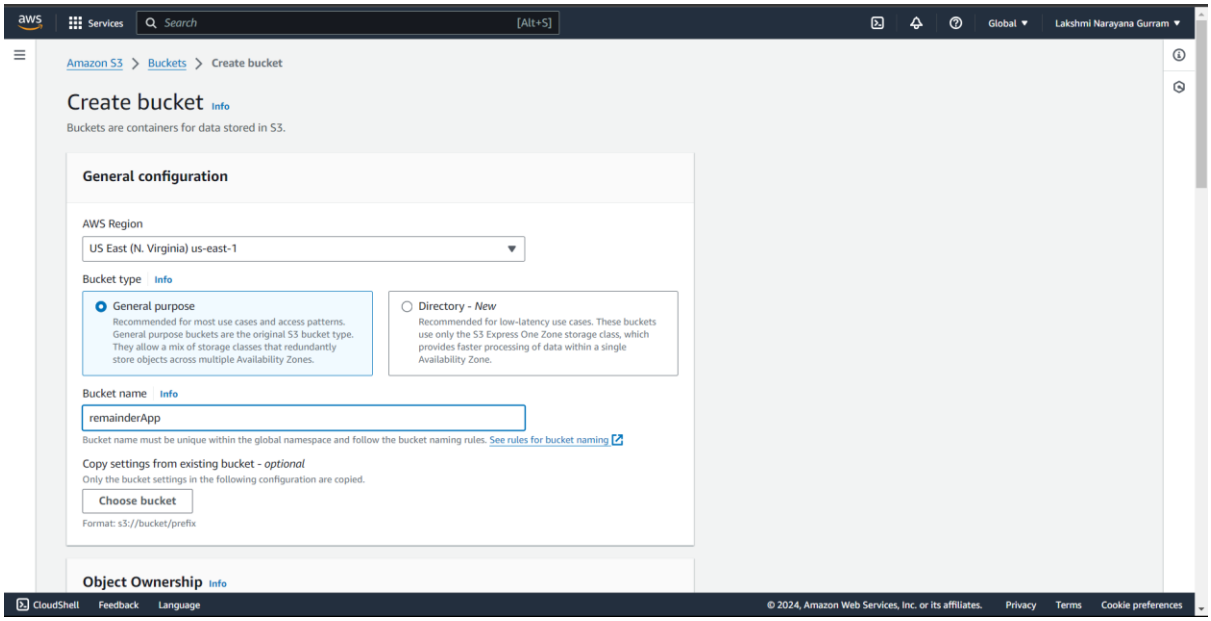
```
print("Received event:", json.dumps(event))

# Check if 'body' attribute is present in the event
if 'body' not in event:
    return {
        "statusCode": 400,
        "headers": {"Access-Control-Allow-Origin": "*"},
        "body": json.dumps({"Status": "Failure", "Reason": "'body' attribute missing in the event"})
    }

try:
    data = json.loads(event['body'])
    data['waitSeconds'] = int(data['waitSeconds'])

    checks = [
        'waitSeconds' in data,
        isinstance(data['waitSeconds'], int),
        'message' in data
    ]

    if False in checks:
        response = {
            "statusCode": 400,
            "headers": {"Access-Control-Allow-Origin": "*"},
            "body": json.dumps({"Status": "Failure", "Reason": "Input failed validation"})
        }
    else:
        sm.start_execution(stateMachineArn=SM_ARN, input=json.dumps(data))
        response = {
            "statusCode": 200,
            "headers": {"Access-Control-Allow-Origin": "*"},
```

```
            "body": json.dumps({"Status": "Success"})
        }
    except Exception as e:
        response = {
            "statusCode": 500,
            "headers": {"Access-Control-Allow-Origin": "*"},
            "body": json.dumps({"Status": "Failure", "Reason": str(e)})
        }

    return response
```

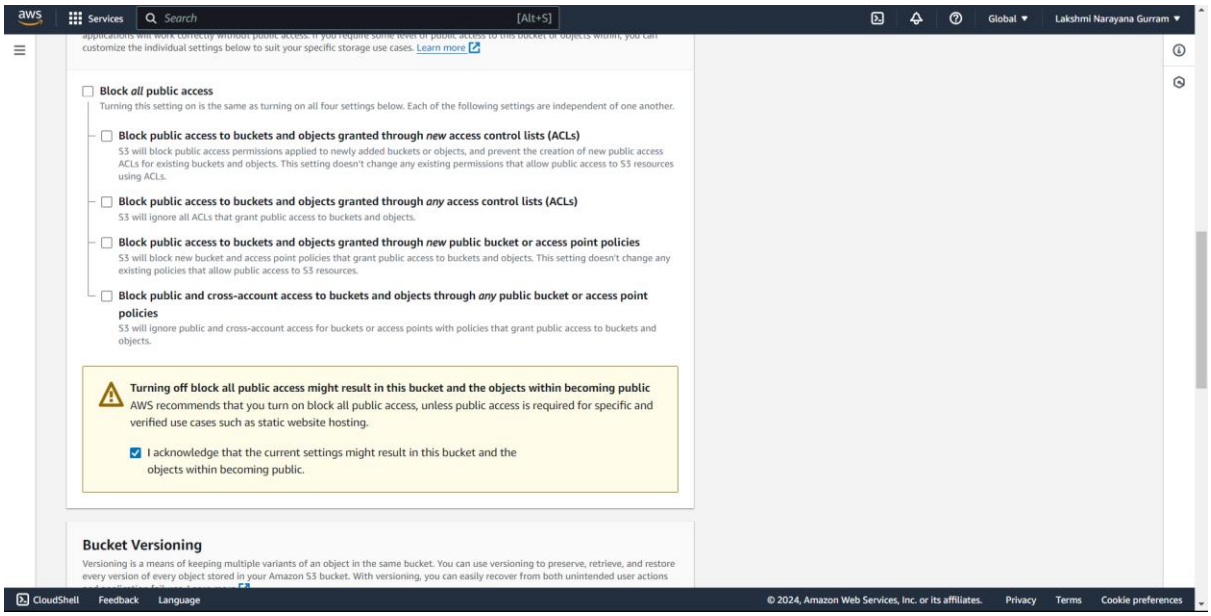
paste and deploy it



Create an S3 bucket



Give it public access

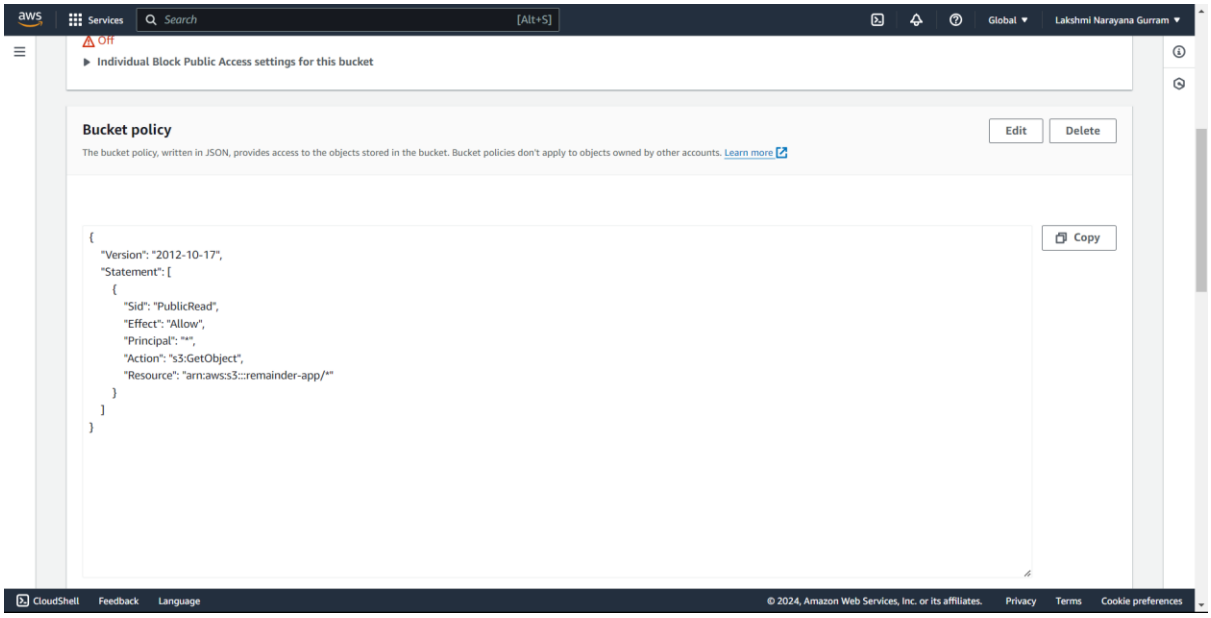


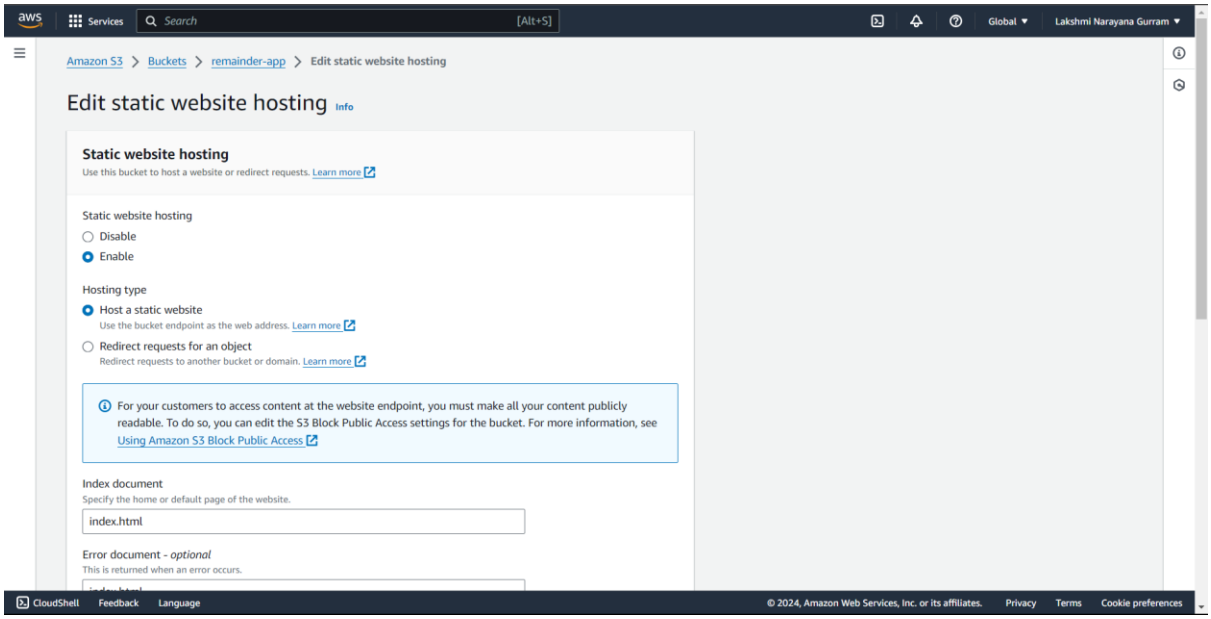
Add bucket policy

Code: {

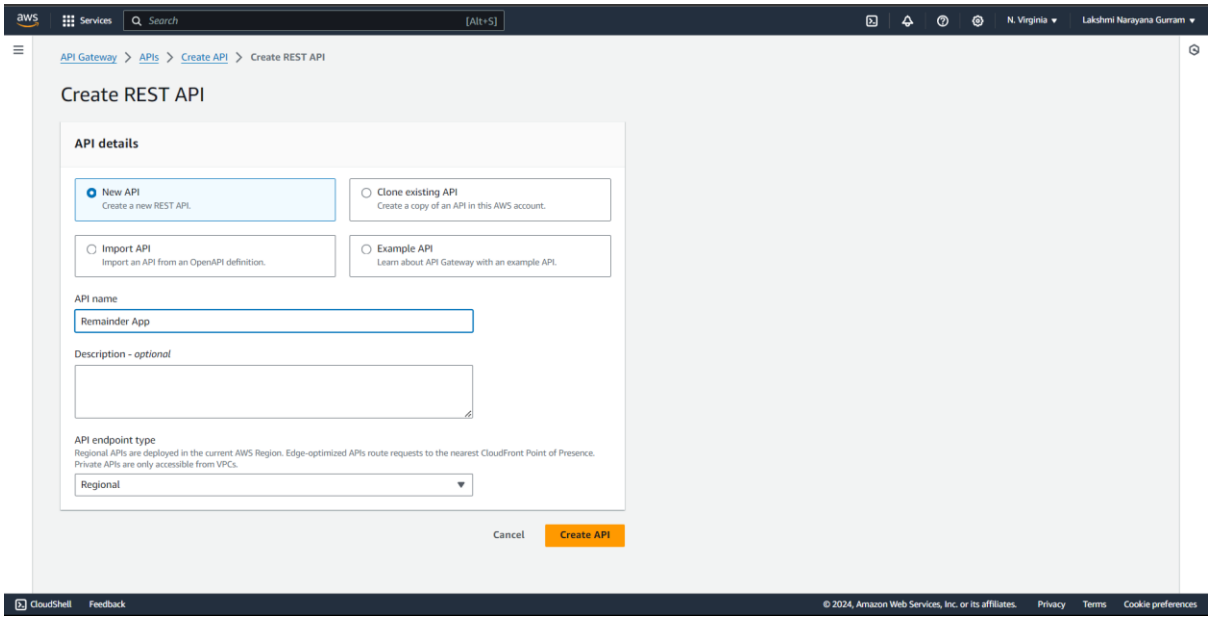
```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
```

```
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::bucketname/*"
      }
    ]
}
```

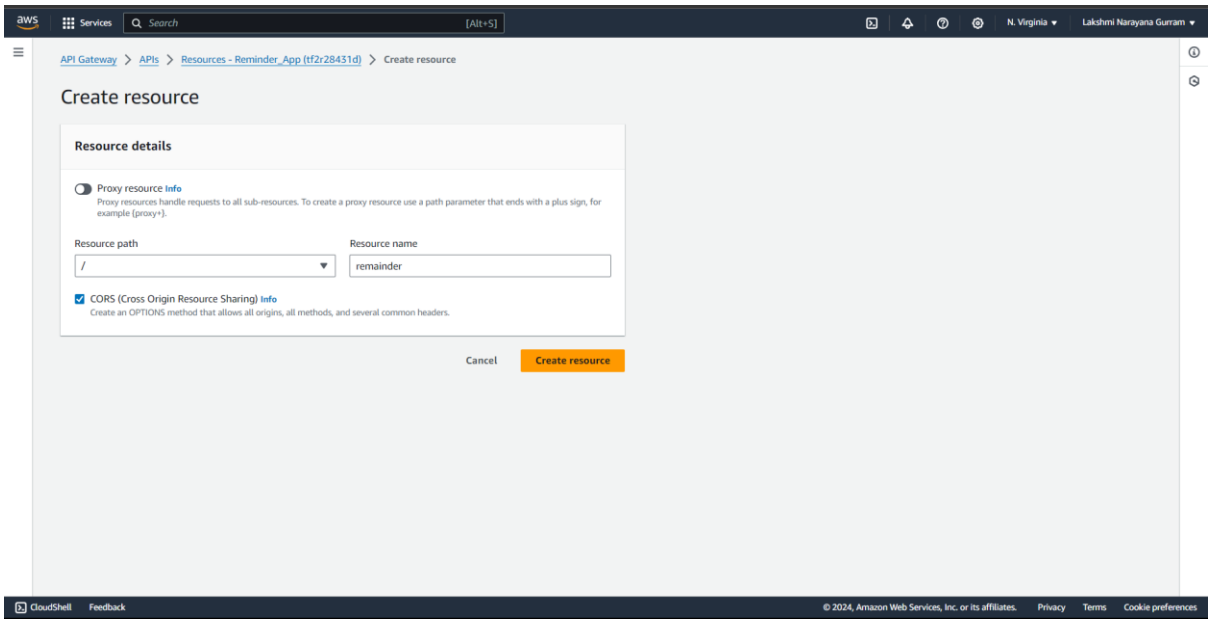




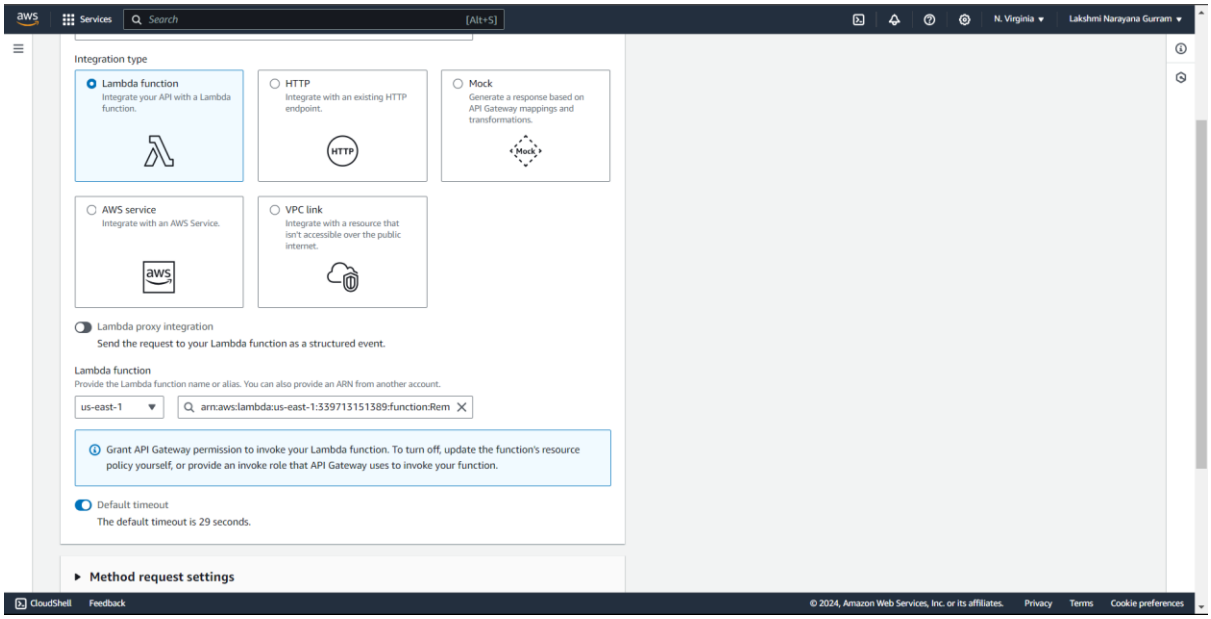
Create an Api gateway



Create a resource in rest api



Create a method post and select lambda
and below select the lambda function



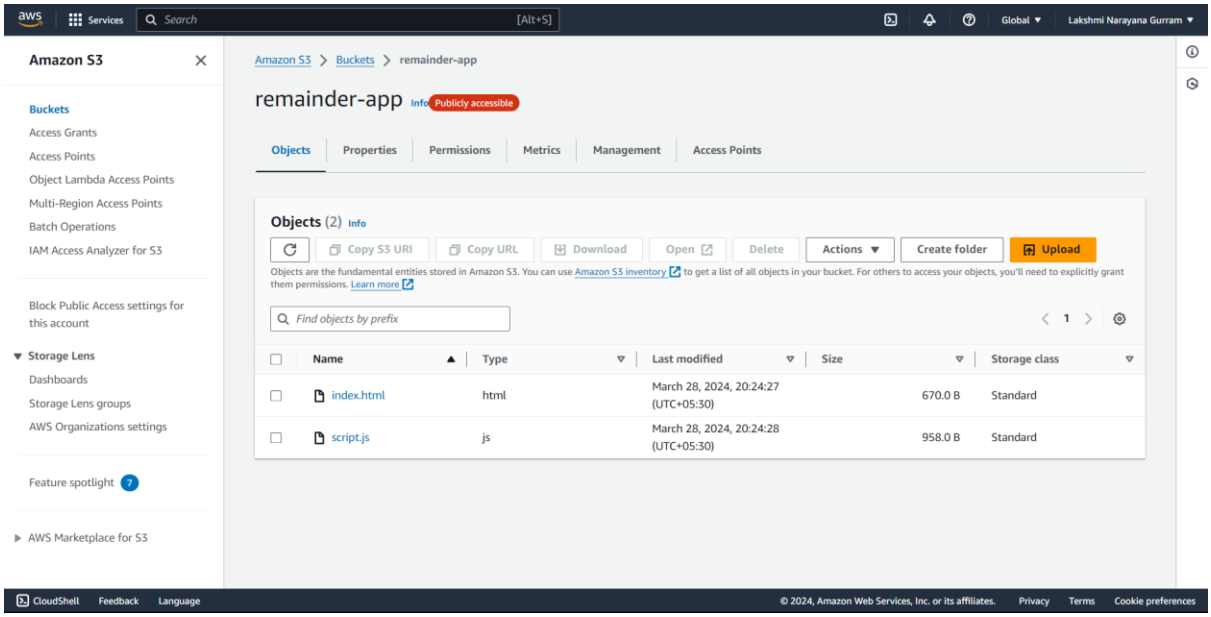
Create a index.html page

```
index.html > ...
    Click here to ask Blackbox to help you code faster
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Send Reminder</title>
7   </head>
8   <body>
9       <h1>Send Reminder</h1>
10      <label for="email">Email:</label>
11  <input type="email" id="email" name="email"><br><br>
12      <label for="message">Message:</label>
13      <input type="text" id="message" name="message"><br><br>
14      <label for="waitSeconds">Wait Seconds:</label>
15      <input type="number" id="waitSeconds" name="waitSeconds"><br><br>
16      <button onclick="sendReminder()">Send Reminder</button>
17
18
19      <script src="scripts.js"></script>
20  </body>
21  </html>
22
```

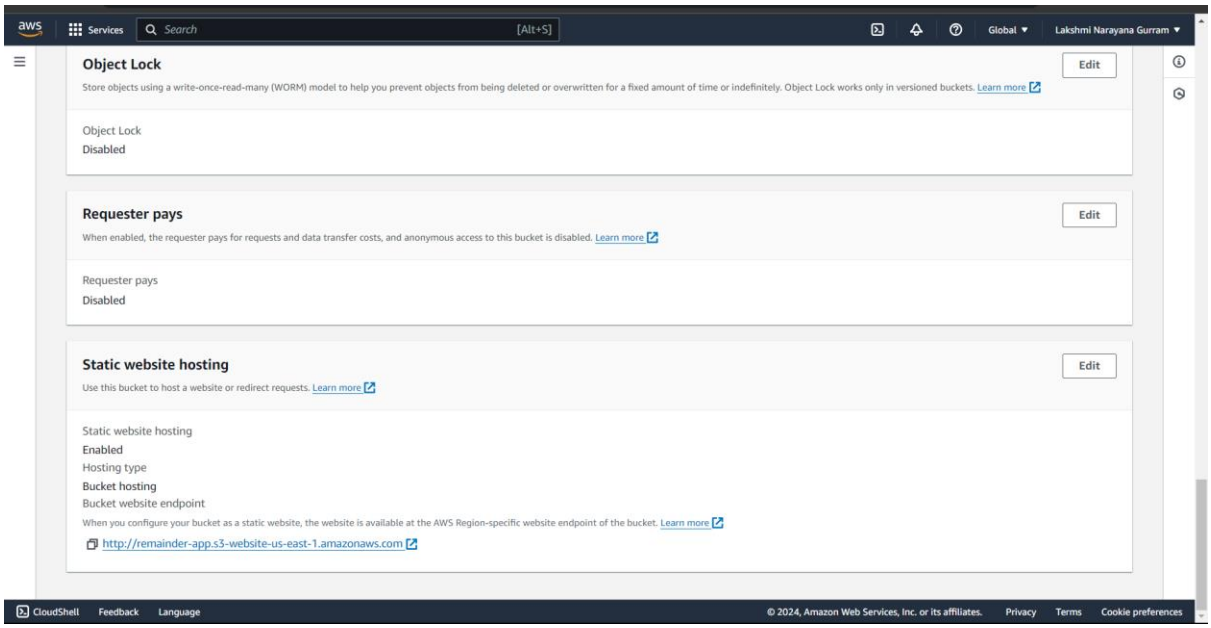
And scripts.js code

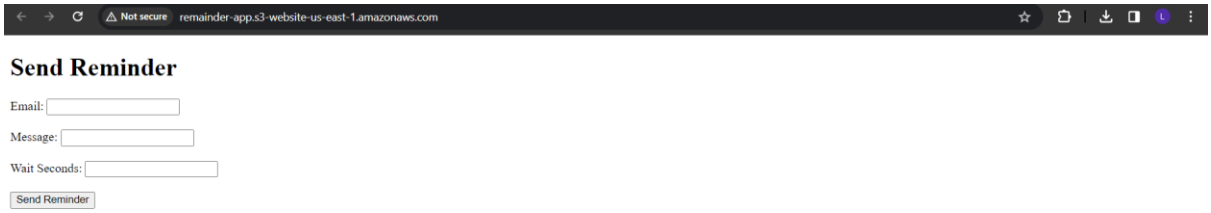
```
Codeium: Options | Test this function | Click here to ask Blackbox to help you code faster
function sendReminder() {
    const email = document.getElementById('email').value;
    const message = document.getElementById('message').value;
    const waitSeconds = document.getElementById('waitSeconds').value;

    const data = {
        email: email,
        message: message,
        waitSeconds: waitSeconds
    };

    fetch('/Remainder', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(data)
    })
    .then(response => {
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        return response.json();
    })
    .then(data => {
        console.log('Success:', data);
        // Handle success response here
    })
    .catch(error => {
        console.error('Error:', error);
        // Handle error here
    });
}
```

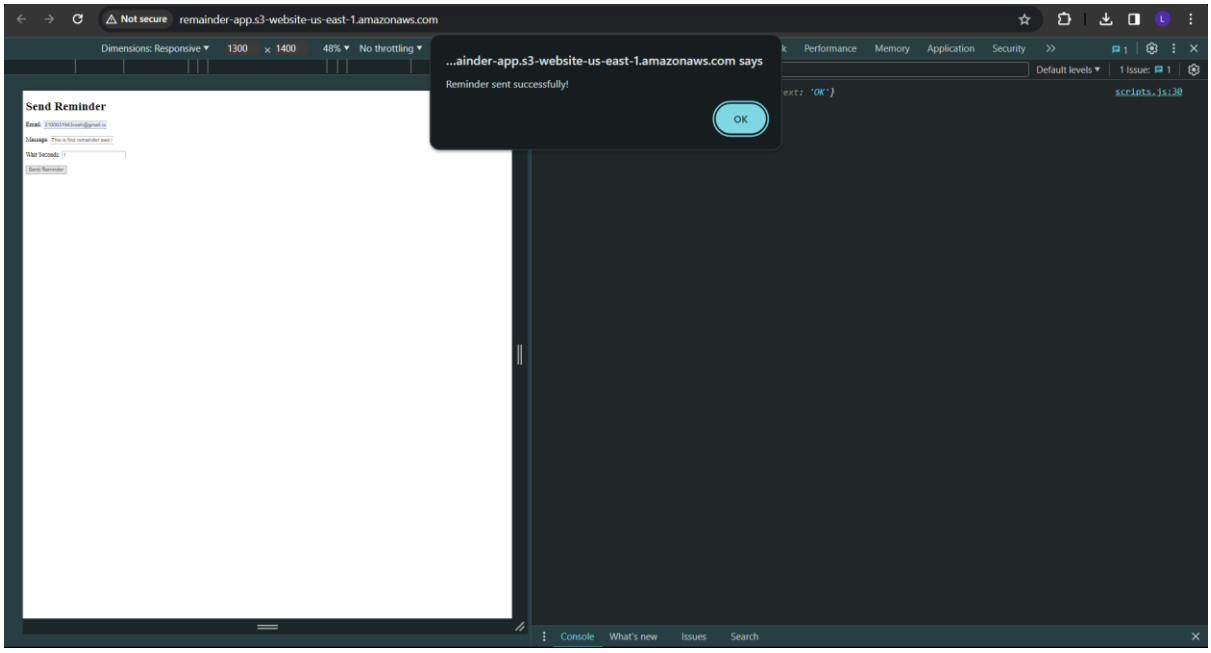
Once create upload them in the previous created s3 bucket



After successfully uploading go to properties ther u will find and static web hosting link click on it




Not secure
remainder-app.s3-website-us-east-1.amazonaws.com
Send Reminder
Email:
Message:
Wait Seconds:
Send Reminder


Not secure
remainder-app.s3-website-us-east-1.amazonaws.com
...ainder-app.s3-website-us-east-1.amazonaws.com says
Reminder sent successfully!
OK
Send Reminder
Console
What's new
Issues
Search

This is the result