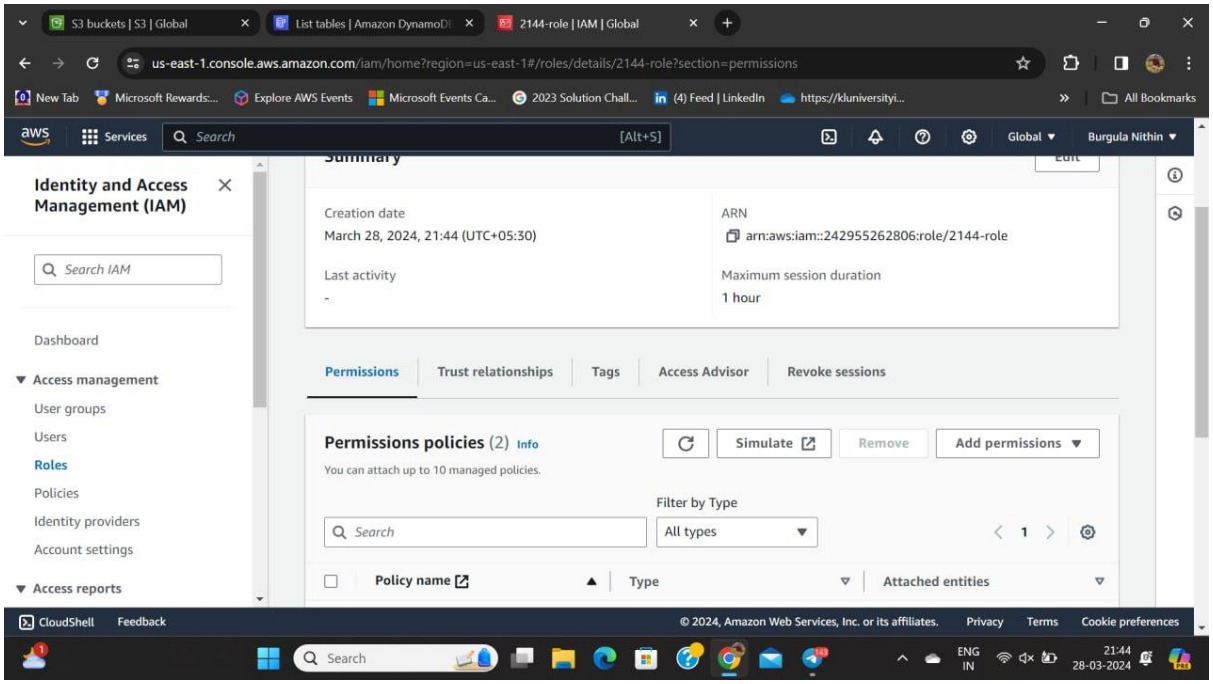
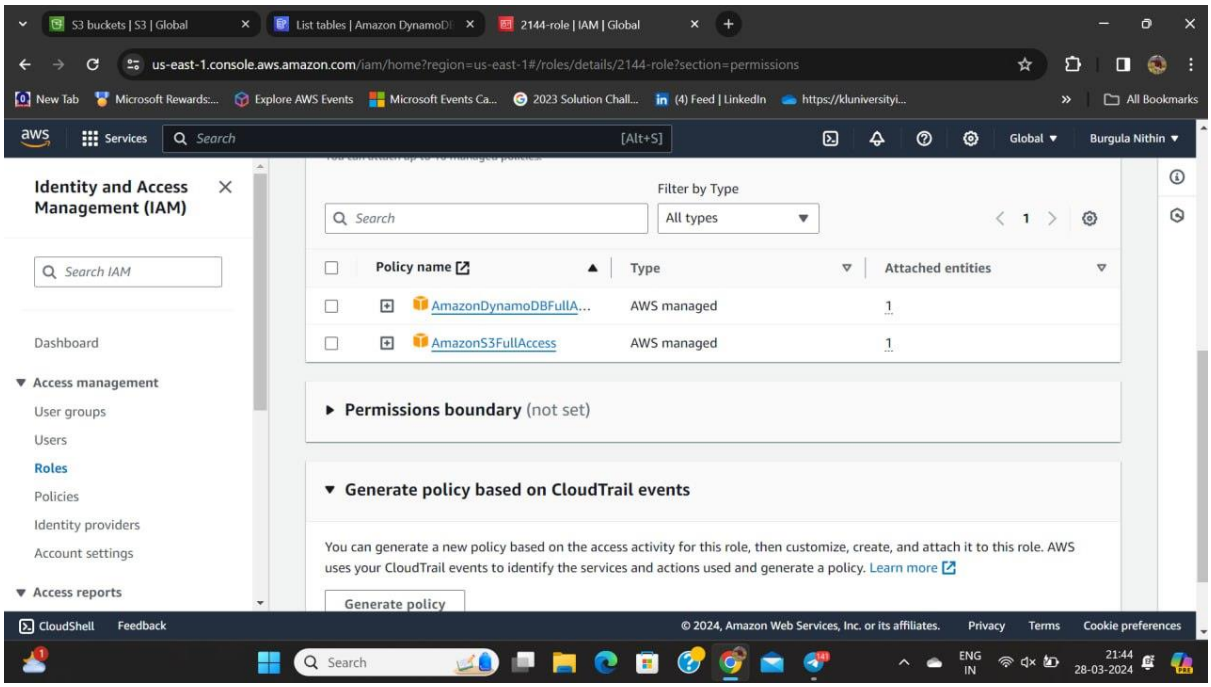
STORAGE CLOUD FILE

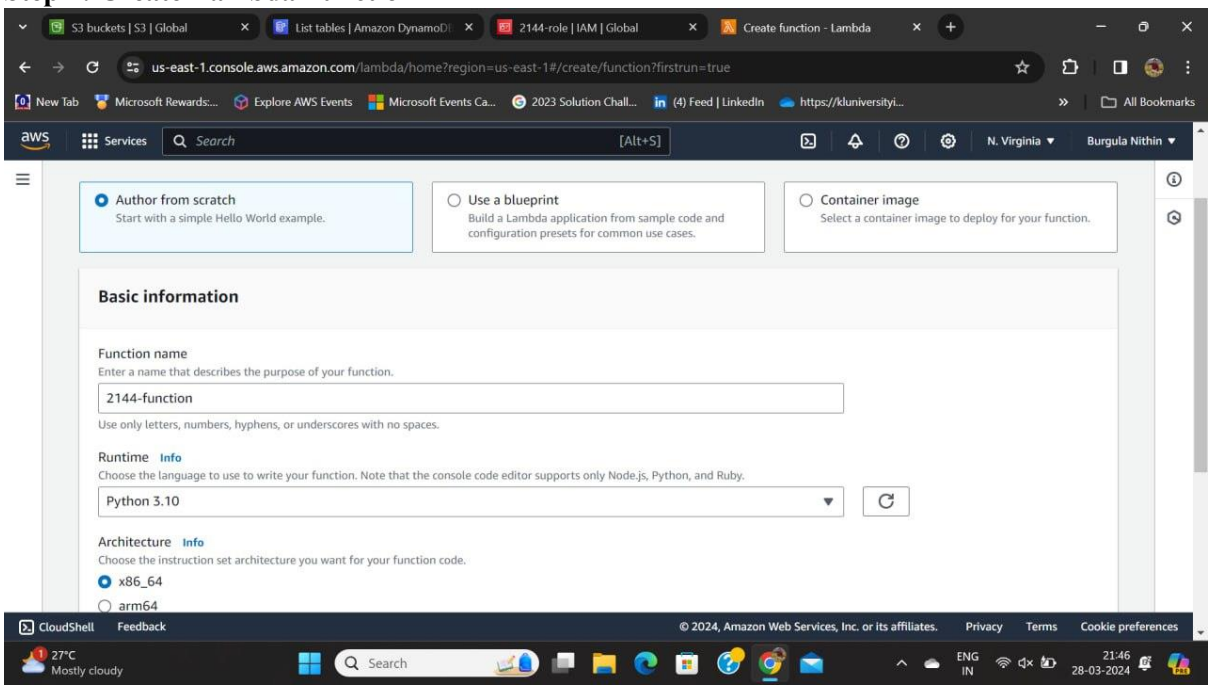
Services Used: S3, DynamoDB, Lambda, API Gateway, IAM Role

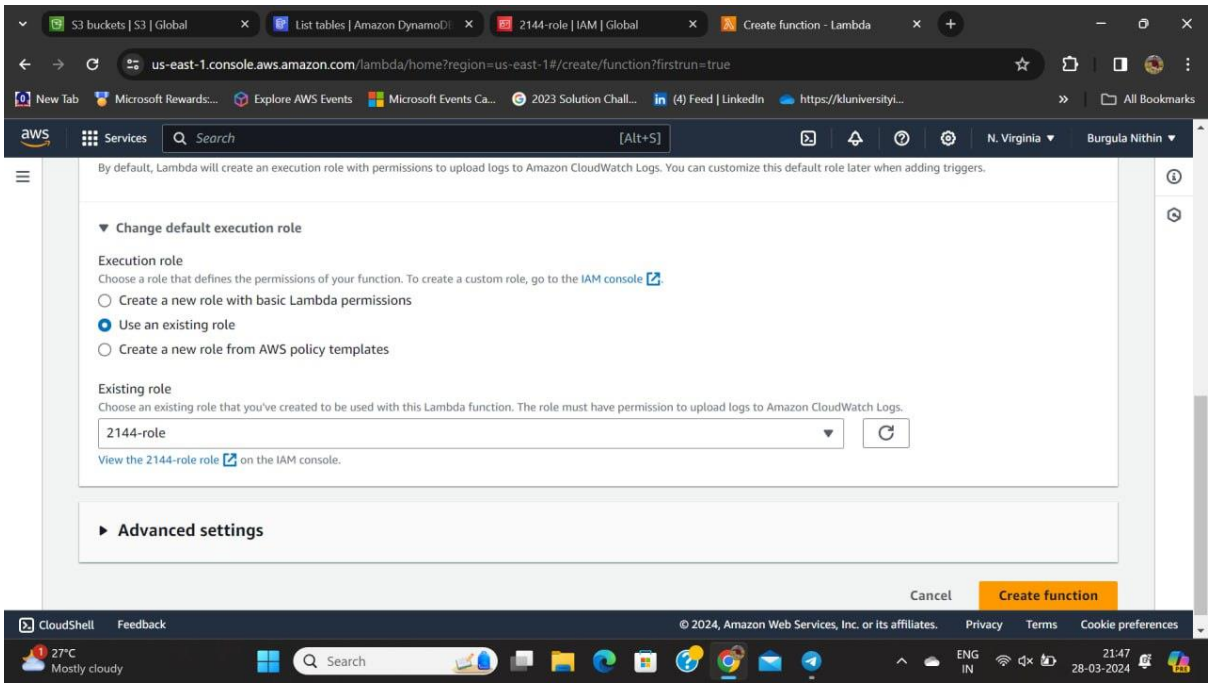
Steps 1: Create IAM Role

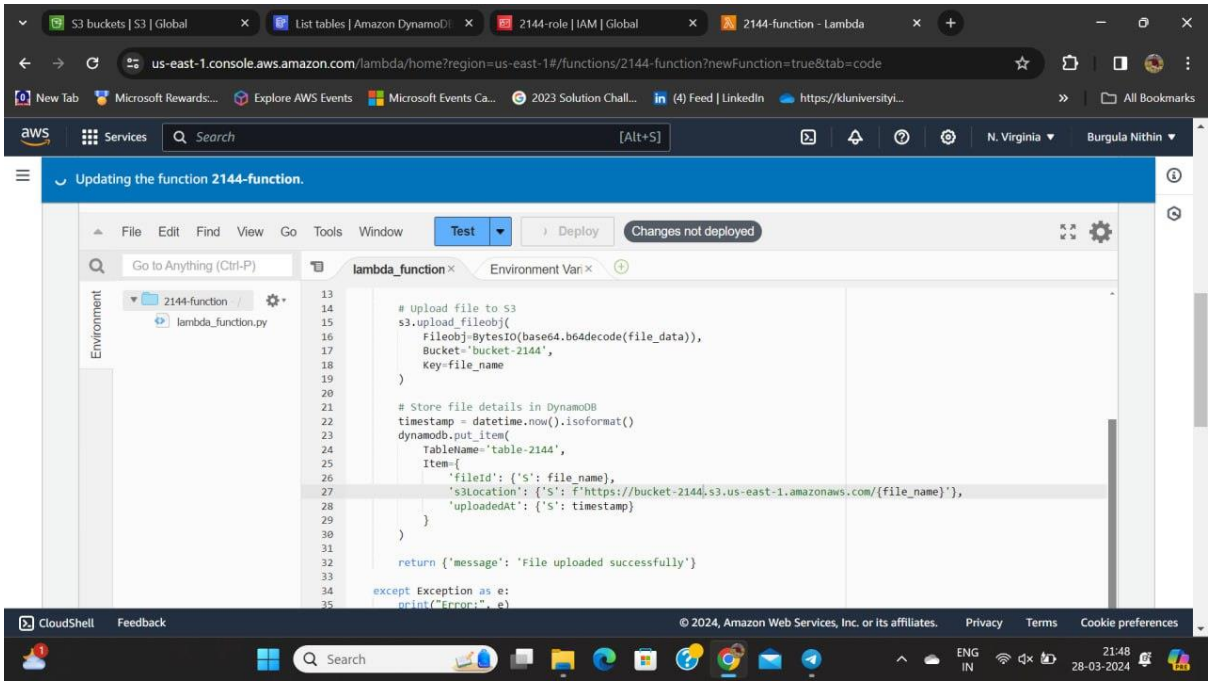




Step 2: Create Lambda Function





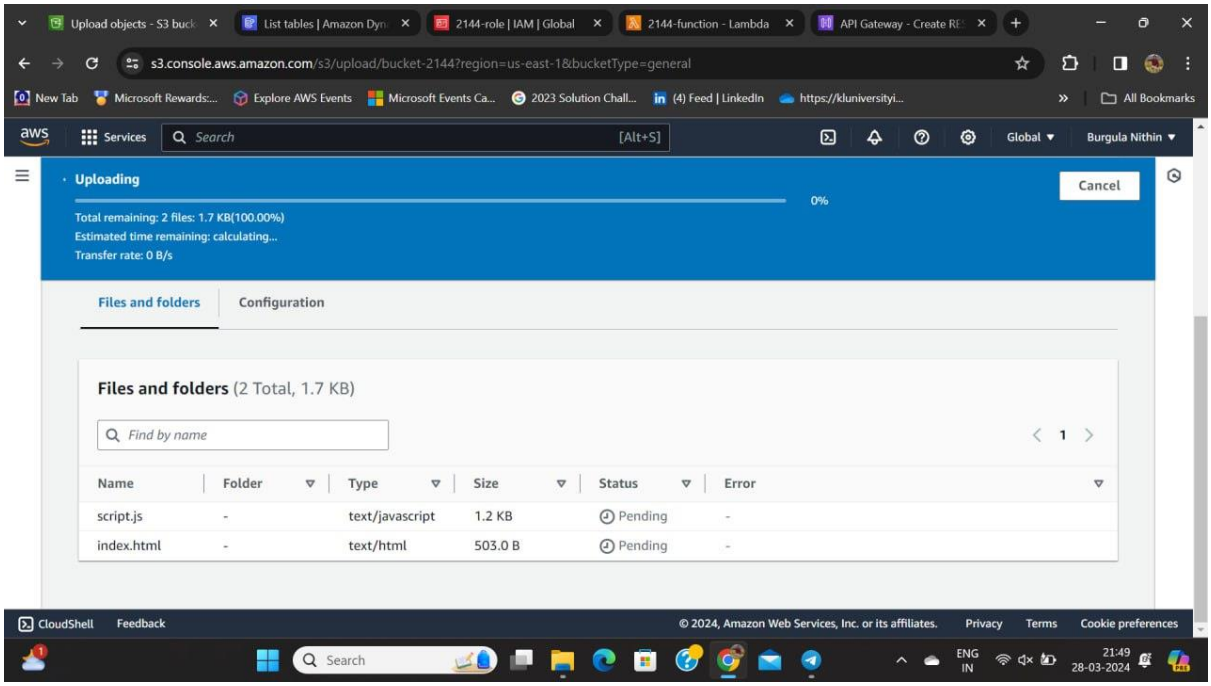


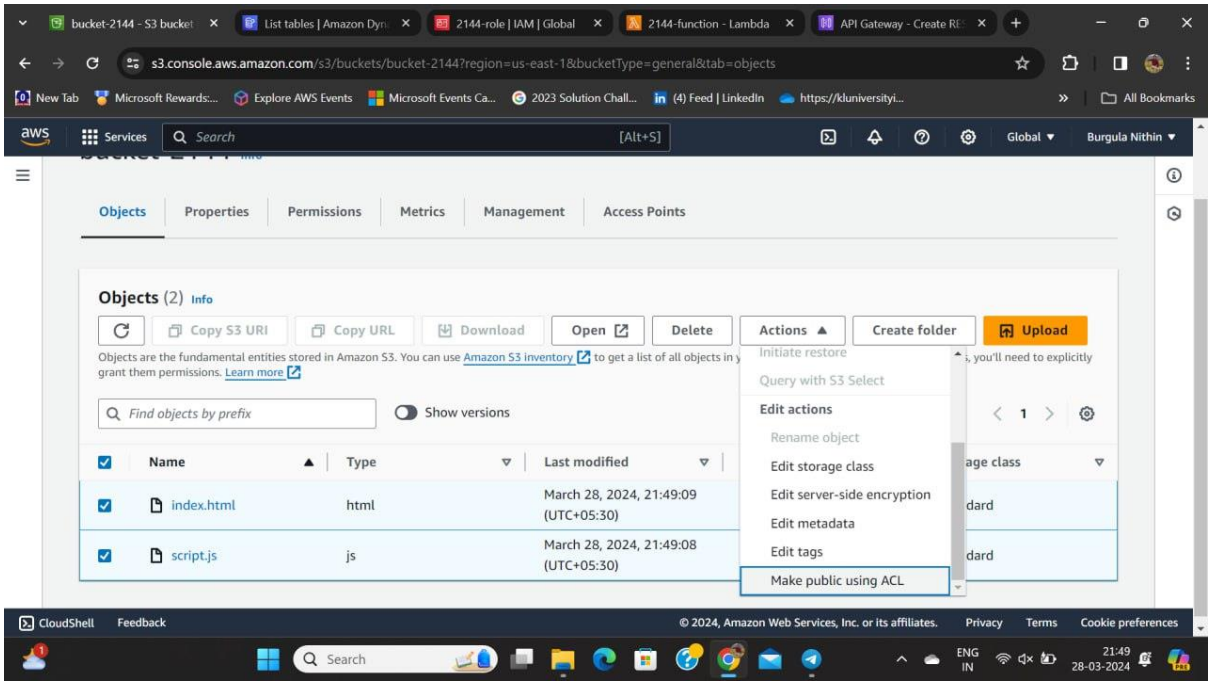
Lambda Function Code:

```
import boto3
import base64
from io import BytesIO
from datetime import datetime

s3 = boto3.client('s3')
dynamodb = boto3.client('dynamodb')

def lambda_handler(event, context):
    try:
        file_name = event['fileName']
        file_data = event['fileData'].split(',')[1]

        # Upload file to S3
        s3.upload_fileobj(
            Fileobj=BytesIO(base64.b64decode(file_data)),
            Bucket='bucket-2144',
            Key=file_name
        )
```

```
    # Store file details in DynamoDB
    timestamp = datetime.now().isoformat()
    dynamodb.put_item(
        TableName='table-2144',
        Item={
            'fileId': {'S': file_name},
            's3Location': {'S': f'https://bucket-2144.s3.us-east-1.amazonaws.com/{file_name}'},
            'uploadedAt': {'S': timestamp}
        }
    )

    return {'message': 'File uploaded successfully'}

except Exception as e:
    print("Error:", e)
    raise Exception('Failed to upload file')
```

Step 3: Upload script.js and index.html





Script.js:

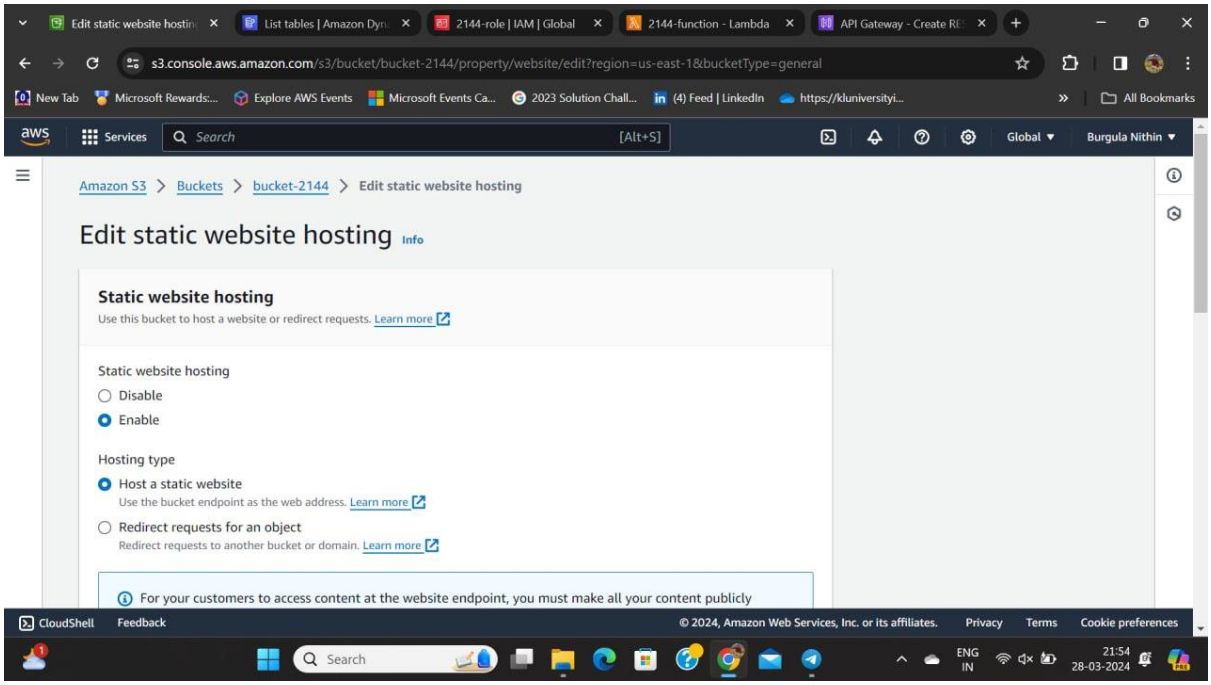
```
function uploadFile() {
  var fileInput = document.getElementById("pdfFile");
  var file = fileInput.files[0];

  var reader = new FileReader();
  reader.onload = function (event) {
    var fileData = event.target.result;
    var params = {
      fileName: file.name,
      fileData: fileData,
    };
    invokeLambdaFunction(params);
  };
  reader.readAsDataURL(file);
}

function invokeLambdaFunction(params) {
  // Replace 'YOUR_API_GATEWAY_URL' with the actual URL of your deployed API
Gateway
  var apiGatewayUrl =
    "https://v8uznb2lak.execute-api.us-east-1.amazonaws.com/test";

  fetch(apiGatewayUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(params),
  })
```

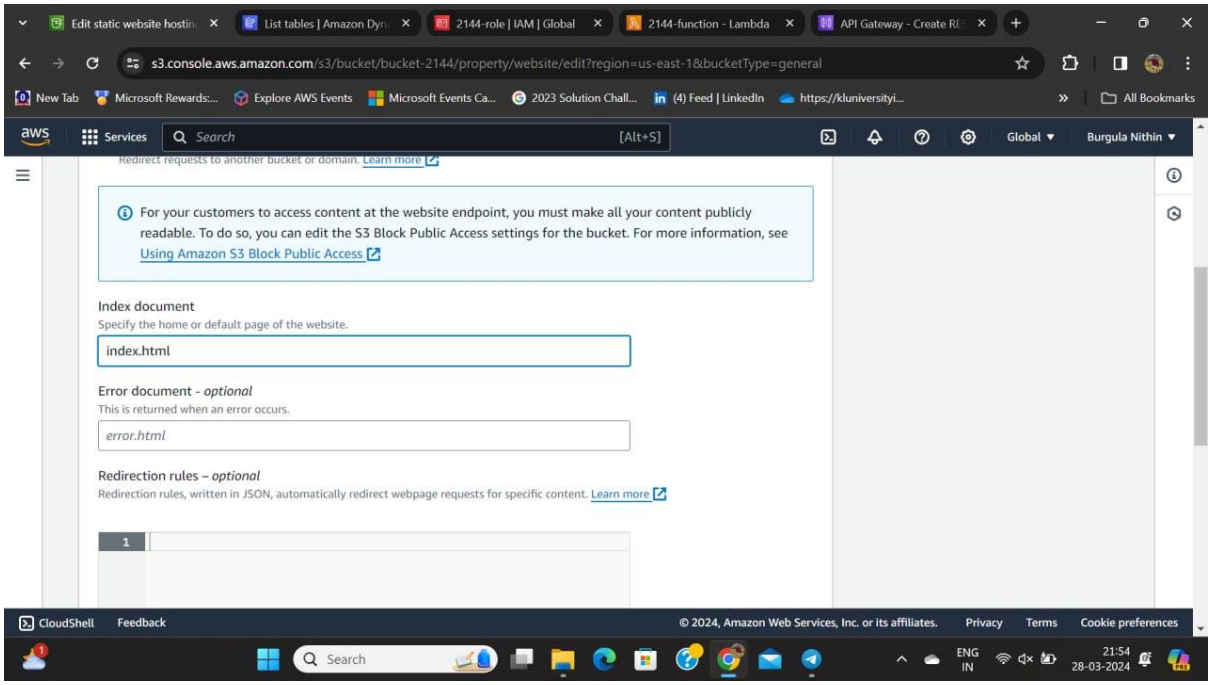
```
    .then((response) => response.json())
    .then((data) => {
      console.log("File uploaded successfully. Response: ", data);
      document.getElementById("message").innerHTML =
        "File uploaded successfully!";
    })
    .catch((error) => {
      console.error("Error uploading file: ", error);
      document.getElementById("message").innerHTML = "Error uploading file!";
    });
}
```

Index.html:

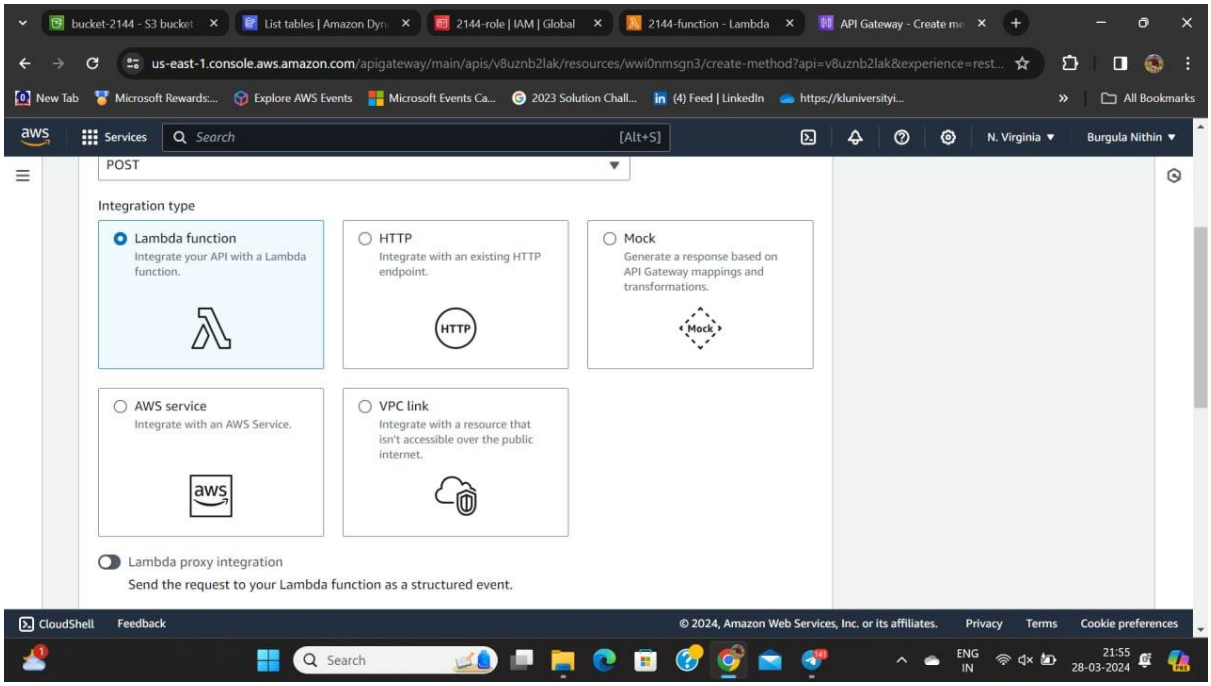
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Cloud File Manager</title>
  <link rel="stylesheet" href="./style.css" />
</head>
<body>
  <header>
    <h1>Cloud File Manager</h1>
    <p>Upload, download, and manage your files securely in the cloud.</p>
  </header>

  <main>
    <section id="upload-section">
      <h2>Upload File</h2>
      <form id="upload-form">
        <input type="file" id="pdfFile" accept=".pdf" required />
        <button type="submit">Upload</button>
      </form>
      <div id="upload-message"></div>
```

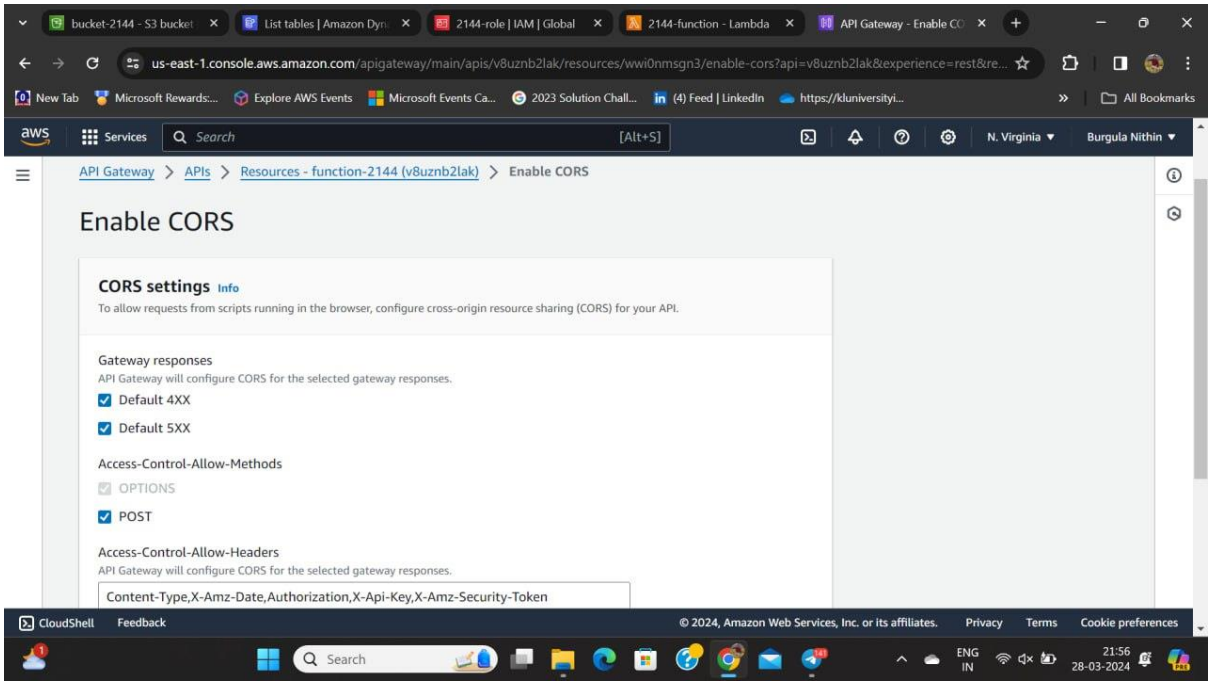
```
  </section>

  <section id="file-list-section">
    <h2>Your Files</h2>
    <ul id="file-list"></ul>
  </section>
</main>

<!-- Add buttons for download and delete actions -->
<section id="file-actions-section">
  <h2>File Actions</h2>
  <div id="file-actions">
    <!-- These buttons will be dynamically populated with file names on page load -->
  </div>
</section>

<script src="https://sdk.amazonaws.com/js/aws-sdk-2.984.0.min.js"></script>
<script src="script.js"></script>
</body>
</html>
```

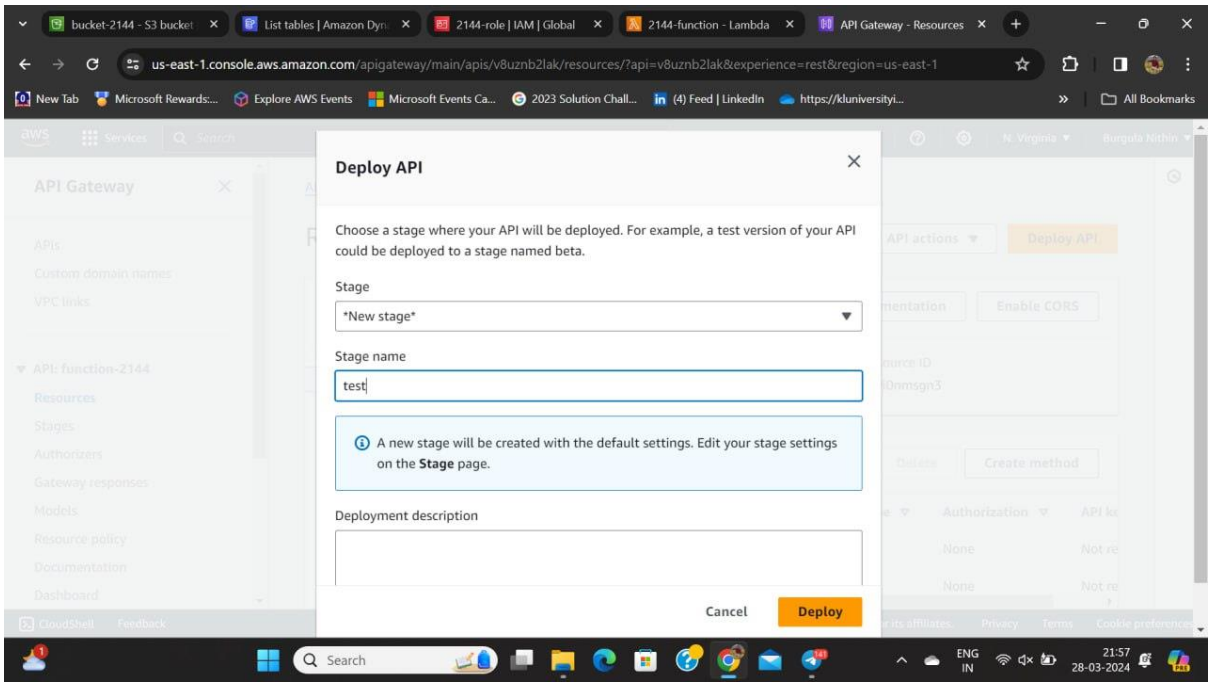


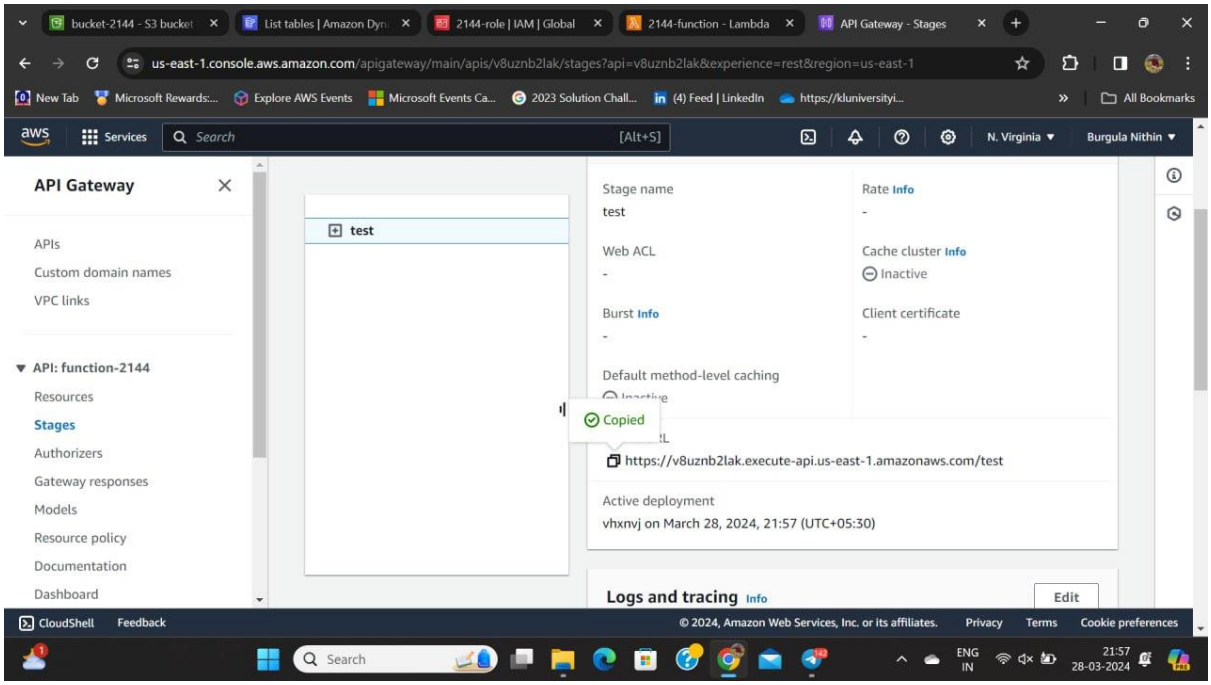


Step 4: Work with Api Gateway

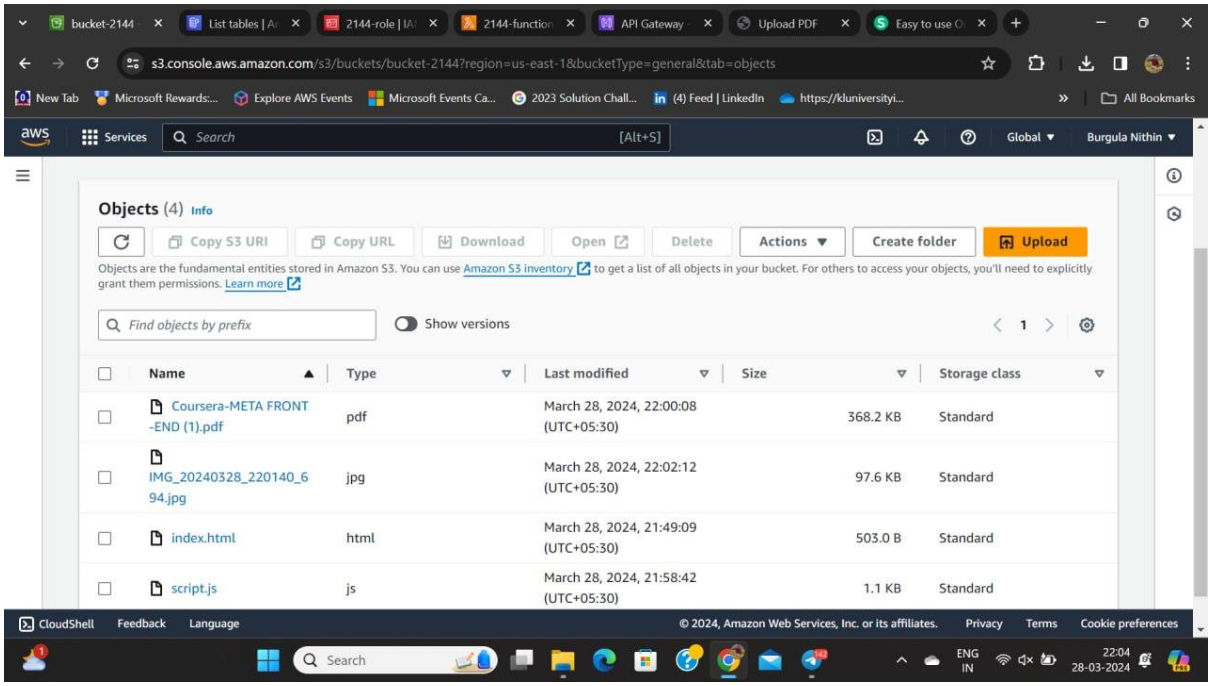



bucket-2144 - S3 bucket
List tables | Amazon Dyn
2144-role | IAM | Global
2144-function - Lambda
API Gateway - Enable CO
us-east-1.console.aws.amazon.com/apigateway/main/apis/v8uznb2lak/resources/wwi0nmsgn3/enable-cors?api=v8uznb2lak&experience=rest&re...
New Tab
Microsoft Rewards:...
Explore AWS Events
Microsoft Events Ca...
2023 Solution Chall...
(4) Feed | LinkedIn
https://kluniversityi...
All Bookmarks
Services
Search
[Alt+S]
N. Virginia
Burgula Nithin
API Gateway > APIs > Resources - function-2144 (v8uznb2lak) > Enable CORS
Enable CORS
CORS settings Info
To allow requests from scripts running in the browser, configure cross-origin resource sharing (CORS) for your API.
Gateway responses
API Gateway will configure CORS for the selected gateway responses.
Default 4XX
Default 5XX
Access-Control-Allow-Methods
OPTIONS
POST
Access-Control-Allow-Headers
API Gateway will configure CORS for the selected gateway responses.
Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
21:56
28-03-2024


bucket-2144 - S3 bucket
List tables | Amazon Dyn
2144-role | IAM | Global
2144-function - Lambda
API Gateway - Resources
us-east-1.console.aws.amazon.com/apigateway/main/apis/v8uznb2lak/resources/?api=v8uznb2lak&experience=rest&region=us-east-1
New Tab
Microsoft Rewards:...
Explore AWS Events
Microsoft Events Ca...
2023 Solution Chall...
(4) Feed | LinkedIn
https://kluniversityi...
All Bookmarks
Deploy API
Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.
Stage
New stage
Stage name
test
A new stage will be created with the default settings. Edit your stage settings on the Stage page.
Deployment description
Cancel
Deploy
Search
ENG
IN
21:57
28-03-2024


API Gateway
APIs
Custom domain names
VPC links
API: function-2144
Resources
Stages
Authorizers
Gateway responses
Models
Resource policy
Documentation
Dashboard
test
Stage name
test
Rate Info
-
Web ACL
-
Cache cluster Info
Inactive
Burst Info
-
Client certificate
-
Default method-level caching
Copied
https://v8uznb2lak.execute-api.us-east-1.amazonaws.com/test
Active deployment
vhxnvj on March 28, 2024, 21:57 (UTC+05:30)
Logs and tracing Info
Edit
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Output:


Objects (4) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
Coursera-META FRONT -END (1).pdf
pdf
March 28, 2024, 22:00:08 (UTC+05:30)
368.2 KB
Standard
IMG_20240328_220140_694.jpg
jpg
March 28, 2024, 22:02:12 (UTC+05:30)
97.6 KB
Standard
index.html
html
March 28, 2024, 21:49:09 (UTC+05:30)
503.0 B
Standard
script.js
js
March 28, 2024, 21:58:42 (UTC+05:30)
1.1 KB
Standard
CloudShell
Feedback
Language
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Using URL files can be uploaded

URL: http://bucket-2144.s3-website-us-east-1.amazonaws.com/