|  |
| --- |
| **Experiment-10: CloudFront**  **Create a cloud front application (feedback), where user has to enter roll No, name, subject (dropdown), rating, and submit button.**  **a) Capture all the details of the user and push into dynamo db.**  **b) Create a CDN for feedback application.**  **c) Create a serverless in ReactJS application and access public domain.**  **Note: This application involves s3 bucket, lambda, cloud front and dynamo db.** |

**Step 1: Start Your AWS Learner Lab**

1. **Go to https://learn.qwiklabs.com.**
2. **Start the lab assigned to you**
3. **Click Start Lab**

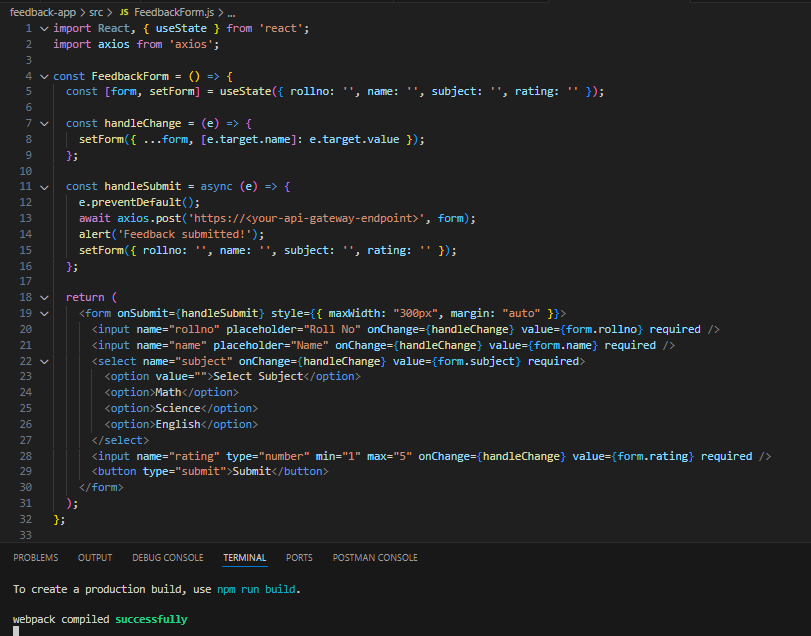
**Step 2: Create Feedback Form App using React:**

**1. Create React app using the command: npx create-react-app feedback**

**2. Create Feedback Component as:**



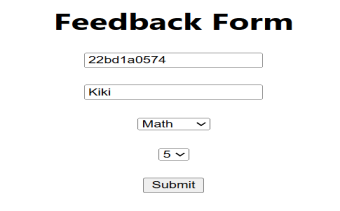
|  |
| --- |
| **Src/App.js**  **import React, { useState } from 'react';**  **import axios from 'axios';**  **import './App.css';**  **function App() {**  **const [rollNo, setRollNo] = useState('');**  **const [name, setName] = useState('');**  **const [subject, setSubject] = useState('Math');**  **const [rating, setRating] = useState('5');**  **const handleSubmit = async (e) => {**  **e.preventDefault();**  **const data = { rollNo, name, subject, rating };**  **try {**  **await axios.post('https://<API\_GATEWAY\_URL>/feedback', data);**  **alert('Feedback submitted!');**  **} catch (error) {**  **console.error(error);**  **alert('Submission failed.');**  **}**  **};**  **return (**  **<div className="App">**  **<h1>Feedback Form</h1>**  **<form onSubmit={handleSubmit}>**  **<input placeholder="Roll No" value={rollNo} onChange={(e) => setRollNo(e.target.value)} required /><br/><br/>**  **<input placeholder="Name" value={name} onChange={(e) => setName(e.target.value)} required /><br/><br/>**  **<select value={subject} onChange={(e) => setSubject(e.target.value)}>**  **<option value="Math">Math</option>**  **<option value="Science">Science</option>**  **<option value="English">English</option>**  **</select><br>**  **</br><br/>**  **<select value={rating} onChange={(e) => setRating(e.target.value)}>**  **{[1, 2, 3, 4, 5].map(n => <option key={n} value={n}>{n}</option>)}**  **</select><br/><br/>**  **<button type="submit">Submit</button>**  **</form>**  **</div>**  **);**  **}**  **export default App;** |



**3. Build the app: npm run build**



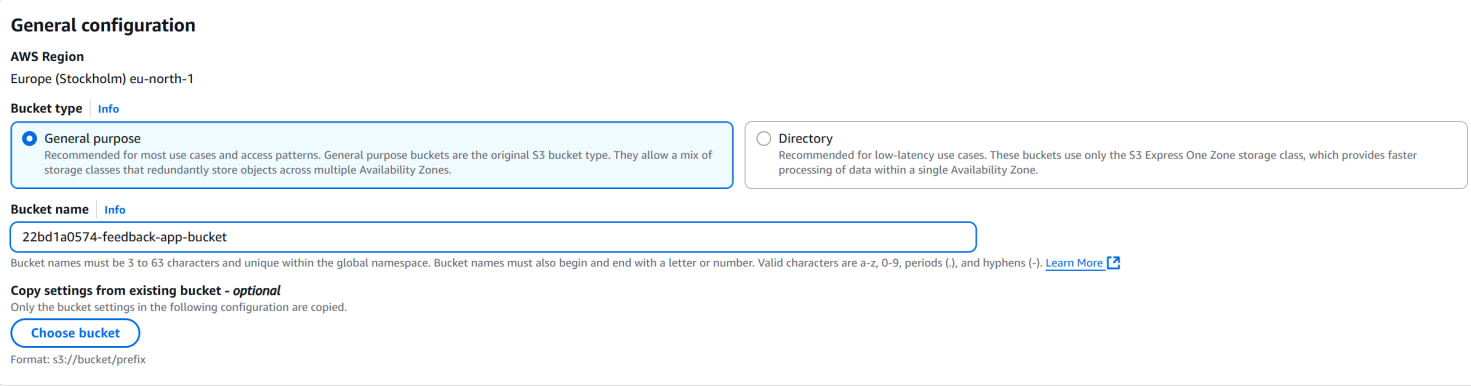


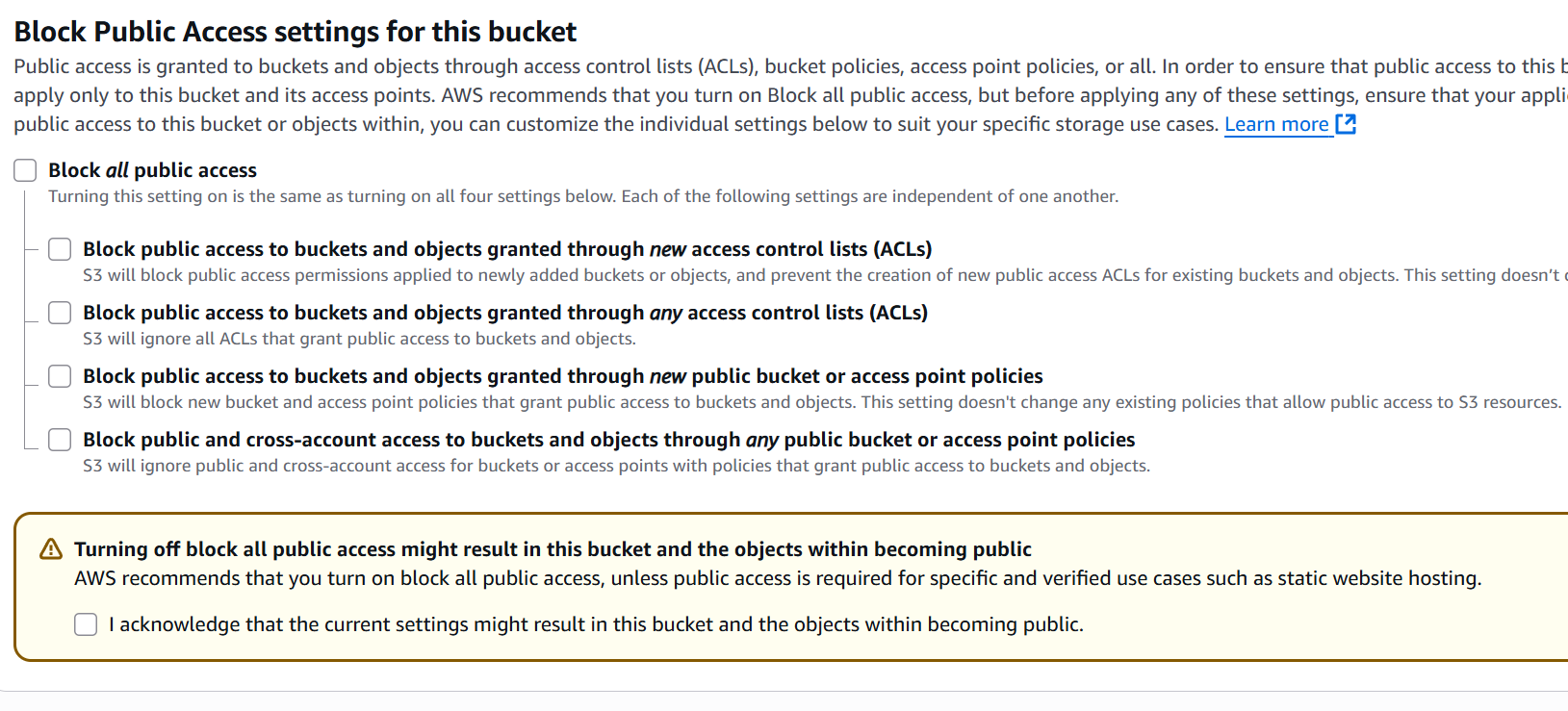


**Step 3: Create & Configure S3 Bucket**

**3.1 Create Bucket**

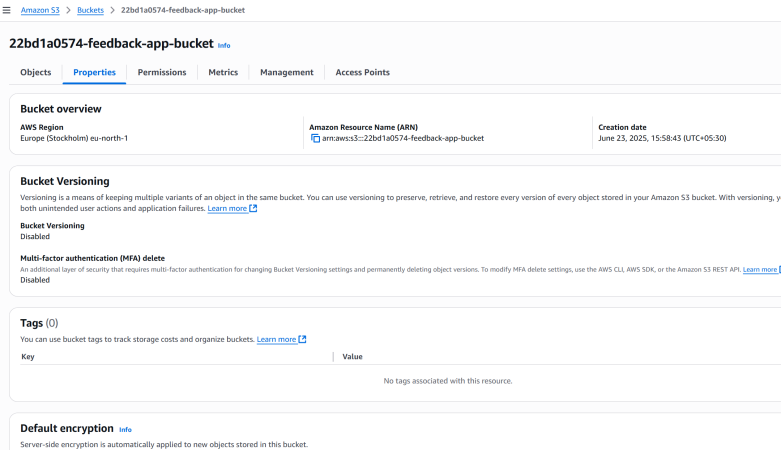
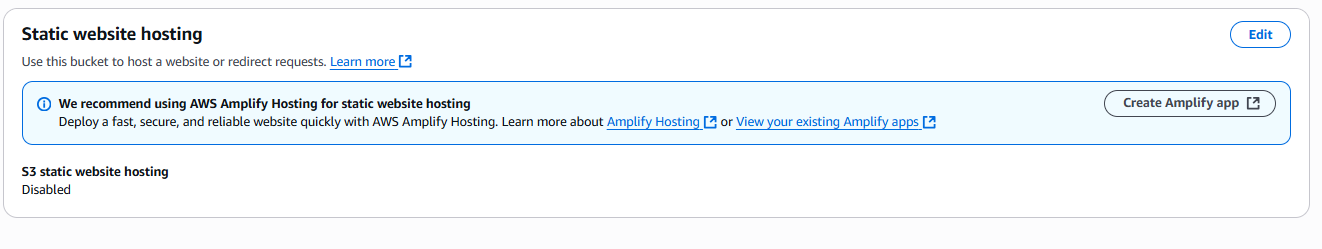
* **Go to S3 > Create bucket**
* **Name: feedback-app-bucket**
* **Region: your region**
* **Uncheck "Block all public access"**

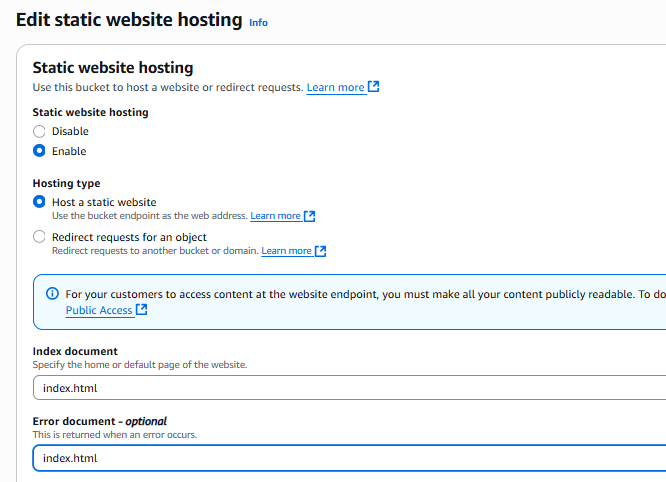




**3.2 Enable Static Website Hosting**

* **Go to Properties > Static website hosting**
* **Enable and set:**
  + **Index document: index.html**
  + **Error document: index.html**



**3.3 Upload React Build**

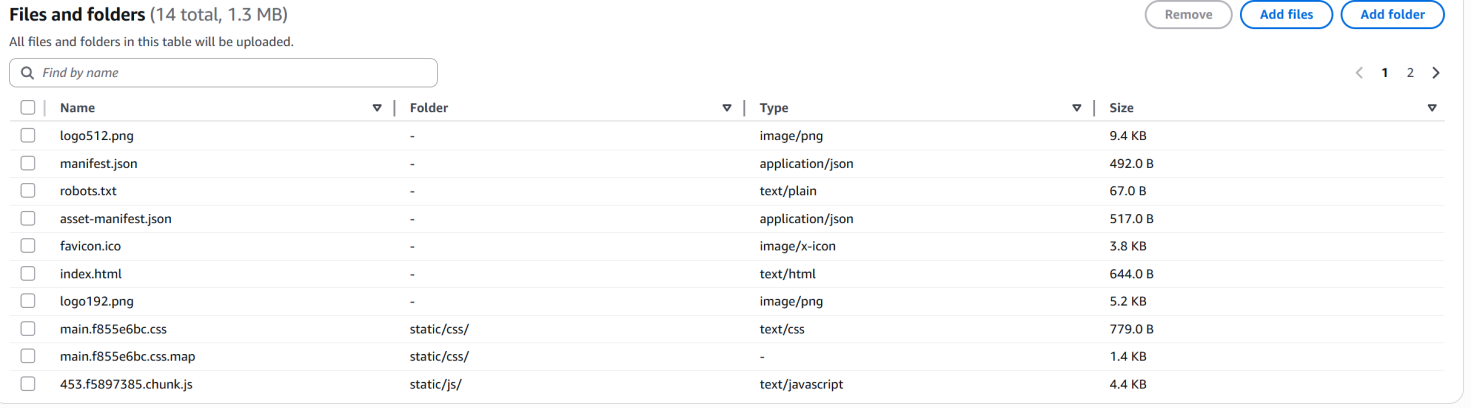
* **Upload all files from the build/ folder.**

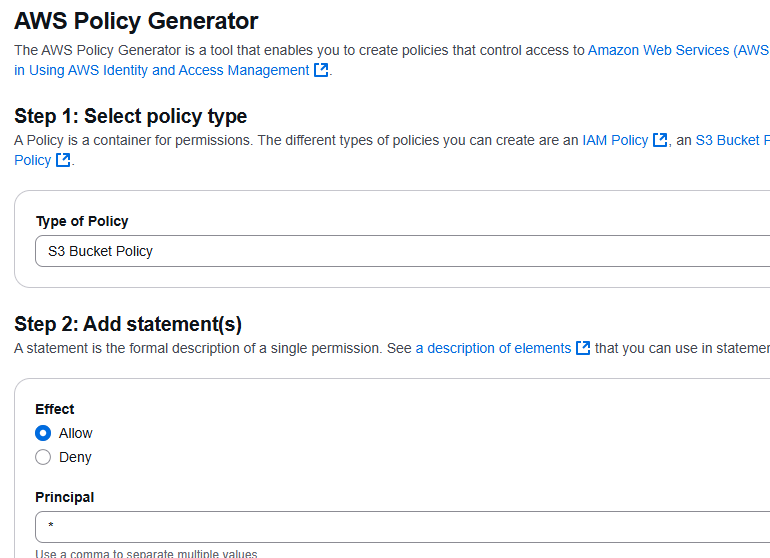
**3.4 Add Bucket Policy (Public Read)**

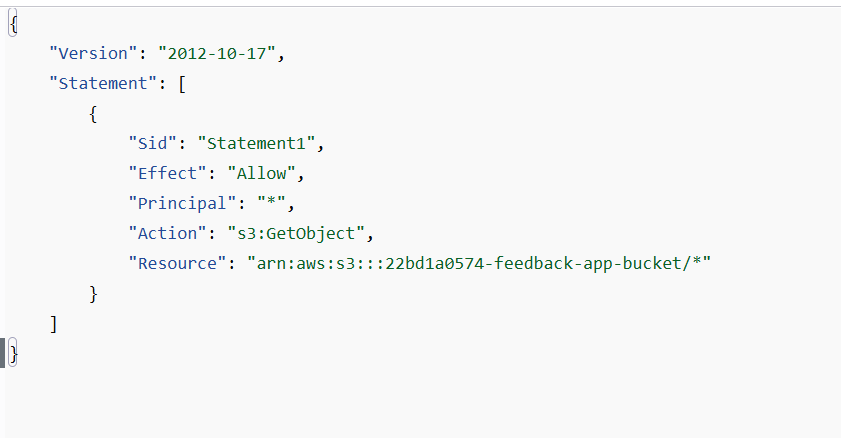
**Go to Permissions > Bucket Policy:**

|  |
| --- |
| **{**  **"Version": "2012-10-17",**  **"Statement": [**  **{**  **"Sid": "Statement1",**  **"Effect": "Allow",**  **"Principal": "\*",**  **"Action": [**  **"s3:GetObject"**  **],**  **"Resource": "arn:aws:s3:::feedback-app-bucket-22bd1a0574/\*"**  **}**  **]**  **}** |





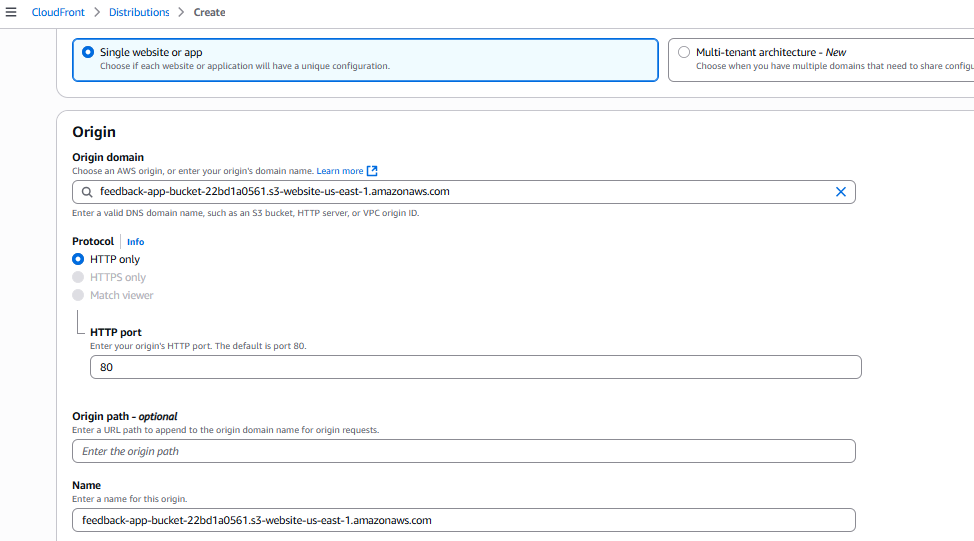


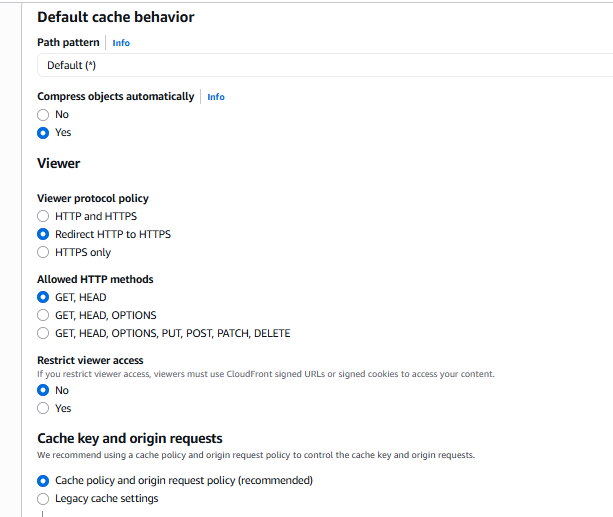


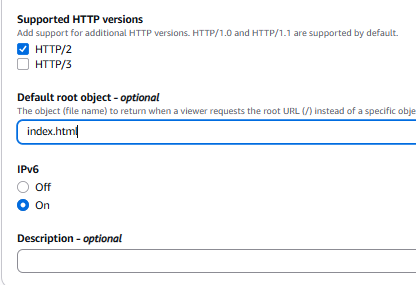
**Step-4: Setup CloudFront CDN**

**4.1 Create CloudFront Distribution**

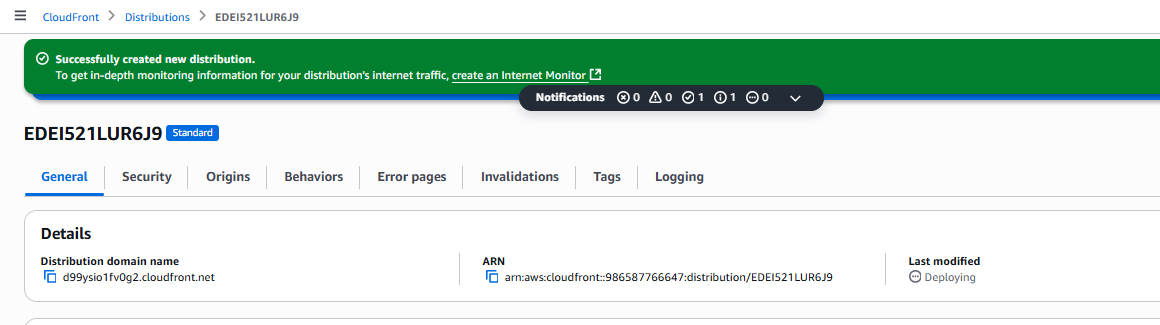
* **Go to CloudFront > Create Distribution**
* **Origin Domain: Select the S3 bucket's static website endpoint**
* **Viewer Protocol Policy: Redirect HTTP to HTTPS**
* **Default root object: index.html**
* **Copy the CloudFront domain name (e.g.,** **https://d1ezjgb2jgubot.cloudfront.net)**





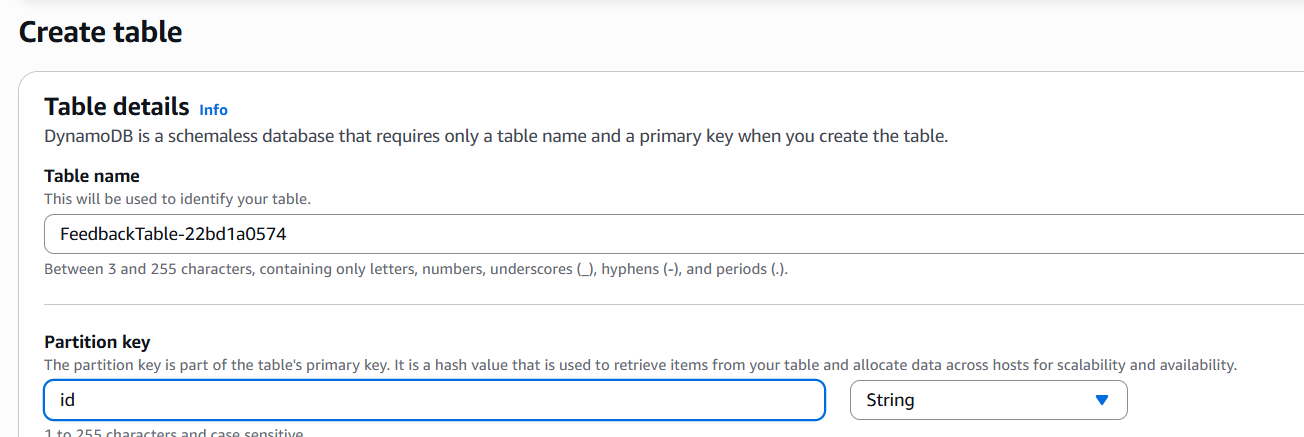


For Web Application Firewall select Do not enable security protection



**Step 5: Create DynamoDB Table**

* **Go to DynamoDB > Create Table**
* **Table name: FeedbackTable-22bd1a0561**
* **Partition key: id (String)**
* **Leave other settings default**



**Step 6: Create Lambda Function:**

**6.1**

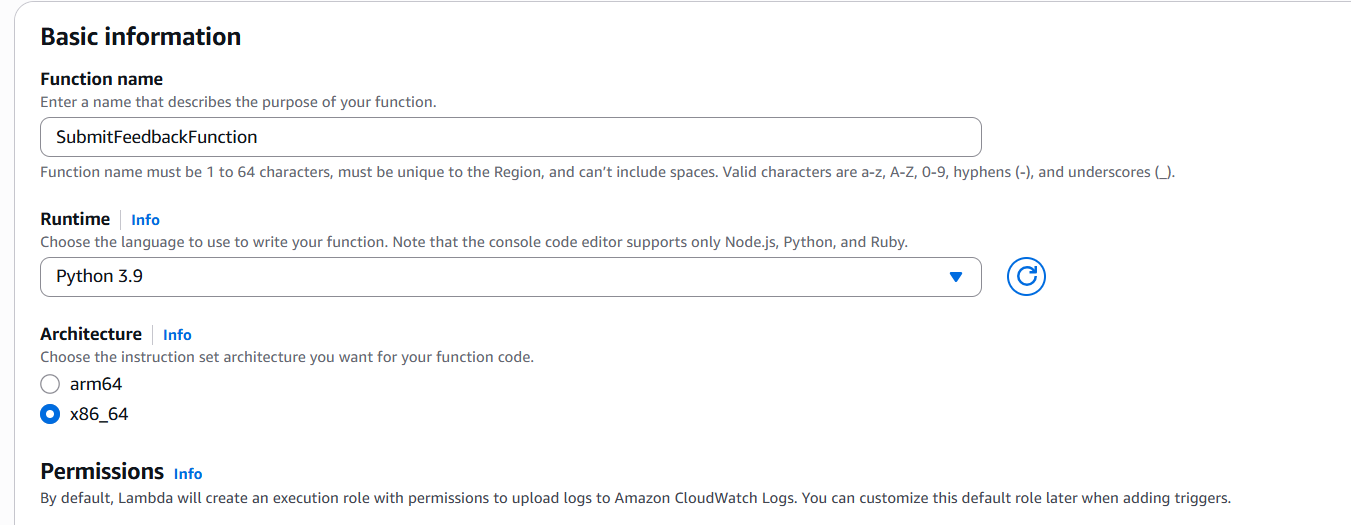
|  |
| --- |
| **Python Code:**  **import json**  **import boto3**  **import uuid**  **dynamodb = boto3.resource('dynamodb')**  **table = dynamodb.Table('FeedbackTable-22bd1a0561')**  **def lambda\_handler(event, context):**  **body = json.loads(event['body'])**    **item = {**  **'id': str(uuid.uuid4()),**  **'rollNo': body['rollNo'],**  **'name': body['name'],**  **'subject': body['subject'],**  **'rating': body['rating']**  **}**    **table.put\_item(Item=item)**  **return {**  **'statusCode': 200,**  **'headers': { 'Access-Control-Allow-Origin': '\*' },**  **'body': json.dumps({ 'message': 'Feedback received!' })**  **}** |

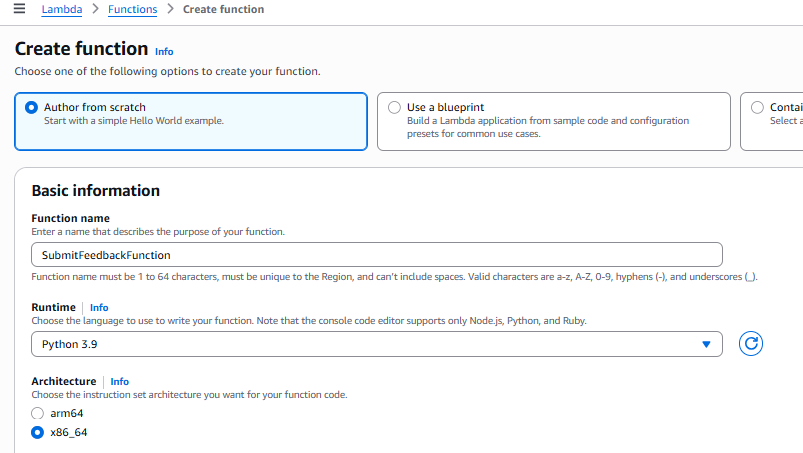
**6.2 Create Lambda**

* **Go to Lambda > Create function**
* **Name: SubmitFeedbackFunction**
* **Runtime: Python 3.9**
* **Paste the code above**

**6.3 Add IAM Permission**

* **Go to Configuration > Permissions > Execution role**
* **Attach AmazonDynamoDBFullAccess policy**







**Step 7: Create API Gateway**

**7.1 Create REST API**

* **Go to API Gateway > Create API > HTTP API**
* **Add Integration: Lambda Function (SubmitFeedbackFunction)**
* **Route: POST /feedback**
* **Stage:Default**

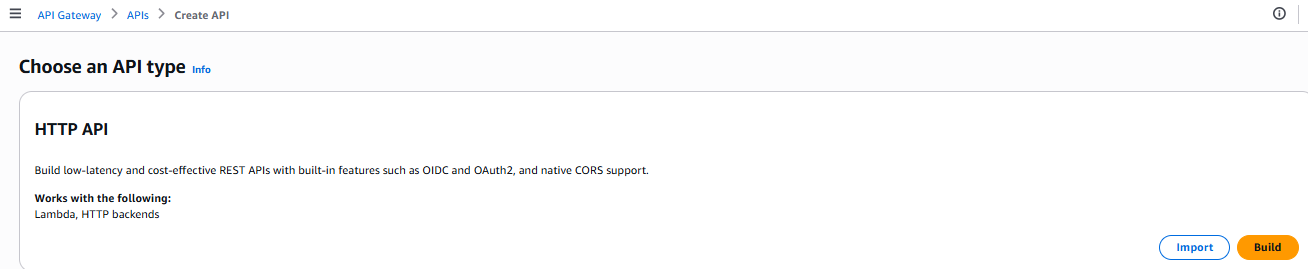
**7.2 Enable CORS**

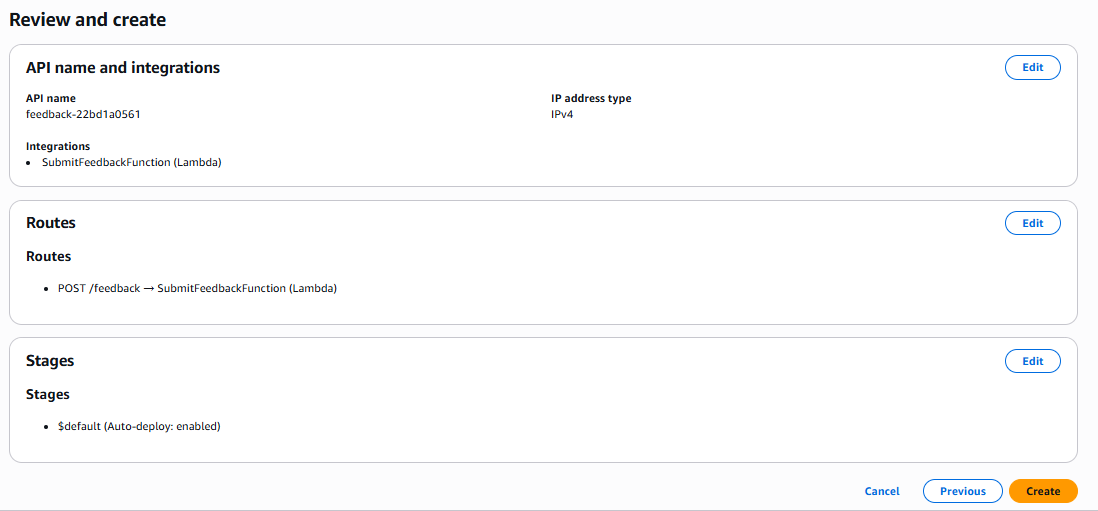
1. **Go to the "Routes" section.**
2. **Select the route, e.g., POST /feedback.**
3. **Click on the "Enable CORS" button.**
4. **Set the following:**
   * **Access-Control-Allow-Headers: Content-Type**
   * **Access-Control-Allow-Methods: POST**
   * **Access-Control-Allow-Origin: \* (or specify your frontend domain like https://d123.cloudfront.net)**

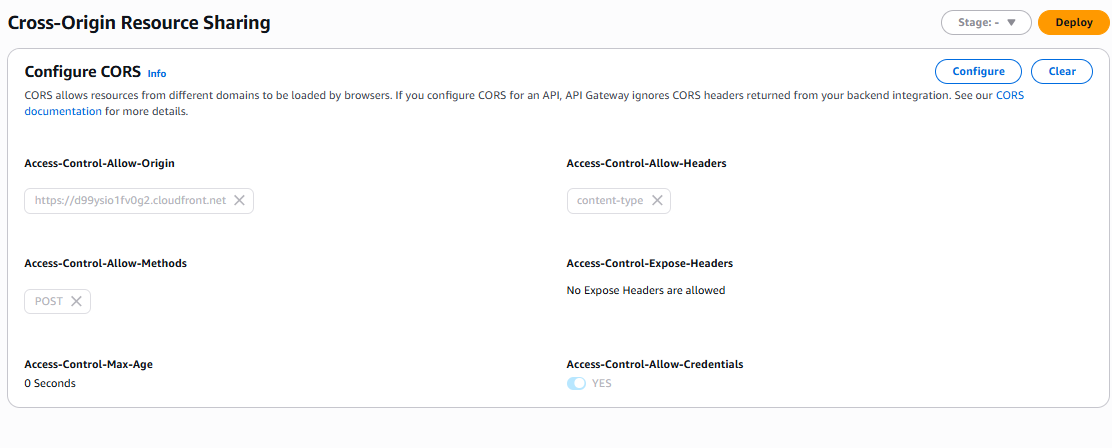
**5. Click "Add CORS configuration".**

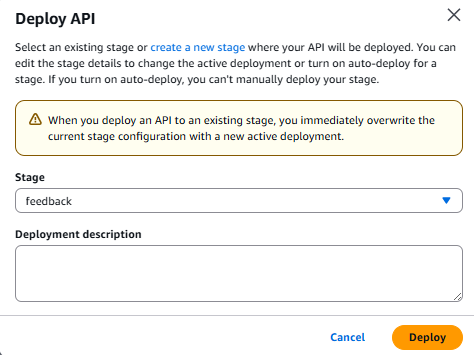
**7.3 Deploy & Copy API URL**

* **Copy the endpoint URL, e.g [https://tfg5hnjvz6.execute-api.eu-north-1.amazonaws.com](https://tfg5hnjvz6.execute-api.eu-north-1.amazonaws.com/" \t "_blank)/feedback**
* **Replace this URL in React code (App.js)**











**Step 8: Redeploy Frontend**

**8.1 Rebuild React with updated API URL**

**npm run build**

**8.2 Reupload build/ files to S3: same as step 3.**

**8.3 Access App via CloudFront URL:** **<https://d1ezjgb2jgubot.cloudfront.net/>**

