SERVERLESS WEB APPLICATION

CREATE S3 BUCKET

1. Create bucket
2. Bucket name – serverless.app
3. Region – Mumbai(ap-south-1)
4. Tags – add tag(key- owner, value- “your own name”)

Click on create bucket

Now bucket is ready

Upload a such file which we have — profile.html and script.js

CREATE CLOUDFRONT

1. Click on create distribution
2. Origin domain – “select amazons3 which we created s3 bucket”
3. Origin access – Origin access control settings(recommended)
4. Click on create control setting – click create
5. Web Application Firewall – Do not enable security protection
6. Click on create distribution

After that created our CLOUDFRONT – we get one popup for “copy policy” click on copy policy then policy has been copied.

Now go back on your s3bucket and paste this policy on “BUCKET POLICY”

1. GO inside the bucket
2. Permissions
3. Bucket policy – edit – “paste policy here”
4. Save changes

Now, Go back again in CLOUDFRONT DISTRIBUTION which you created

1. Go inside the distribution
2. General
3. Settings – (edit)
4. Default root object – profile.html
5. Save changes
6. Distribution domain name (link) – copy the link
7. Paste on the browser
8. Hit enter



You get this page.   
Now our website is successfully loaded

Create DynamoDb table

1. Create table
2. Table name – employeeProfile (exact same name no other name used)
3. Partition key – empId (exact same name no other name used)
4. Tags – add tag (key – owner , value – “your own name”)
5. CREATE TABLE

CREATE IAM ROLE – POLICY – DYNAMODB FULL ACCESS , DYNAMODB READ ONLY ACESS

CREATE LAMBDA FUNCTION

1. Create function
2. Function name – getemployee
3. Runtime – Python 3.11
4. Change default execution role
5. Use existing role
6. Select role which you had created
7. Create function

Created first function   
now copy code from below and paste it on CODE SOURCE

import json

import boto3

def lambda\_handler(event, context):

dynamodb = boto3.resource('dynamodb', region\_name='ap-south-1')

table = dynamodb.Table('employeeProfile')

response = table.scan()

data = response['Items']

while 'LastEvaluatedKey' in response:

response = table.scan(ExclusiveStartKey=response['LastEvaluatedKey'])

data.extend(response['Items'])

return data

1. DEPLOY
2. CLICK ON TEST
3. CREATE EVENT
4. SAVE
5. NOW TEST THE CODE IS RUNNABLE OR NOT.
6. STATUS - SUCCESSFUL

Now create SECOND FUNCTION

1. Name – insertemployee
2. All steps same as it is from first function
3. Create function
4. Now copy the code from below
5. Paste it on CODE SOURCE

import json

import boto3

# create a DynamoDB object using the AWS SDK

dynamodb = boto3.resource('dynamodb')

# use the DynamoDB object to select our table

table = dynamodb.Table('employeeProfile')

# define the handler function that the Lambda service will use as an entry point

def lambda\_handler(event, context):

# extract values from the event object we got from the Lambda service and store in a variable

firstname = event['empFirstName']

id=event['empId']

lastname=event['empLastName']

age=event['empAge']

# write name and time to the DynamoDB table using the object we instantiated and save response in a variable

response = table.put\_item(

Item={

'empId': id,

'empAge':age,

'empFirstName':firstname,

'empLastName':lastname

})

# return a properly formatted JSON object

return {

'statusCode': 200,

'body': json.dumps('Hello from Lambda, ' + firstname)

}

1. DEPLOY
2. CREATE EVENT

Copy the code from below and paste it in EVENT JSON

{

"empId": "1001",

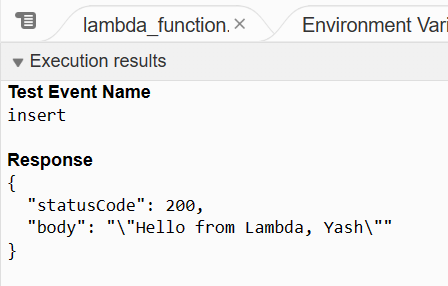
"empAge":"20",

"empFirstName":"Yash",

"empLastName":"Verma"

}

1. SAVE
2. NOW CLICK ON TEST

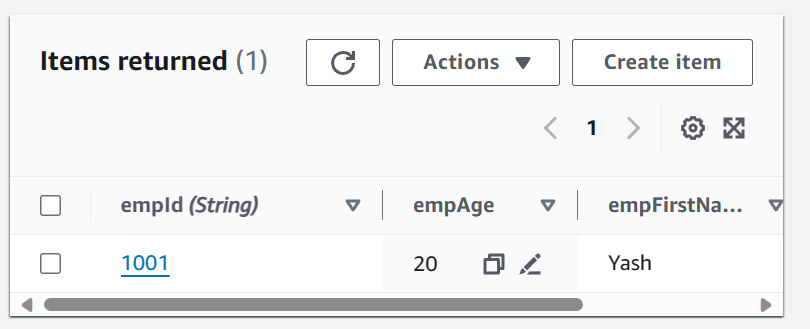


WO GET THIS TYPE OF RESPONSE

THIS IS SAMPLE ENTRY TEST IN DYNAMODB AND THIS ENTRY HAS BEEN SAVE IN DYNAMODB TABLE

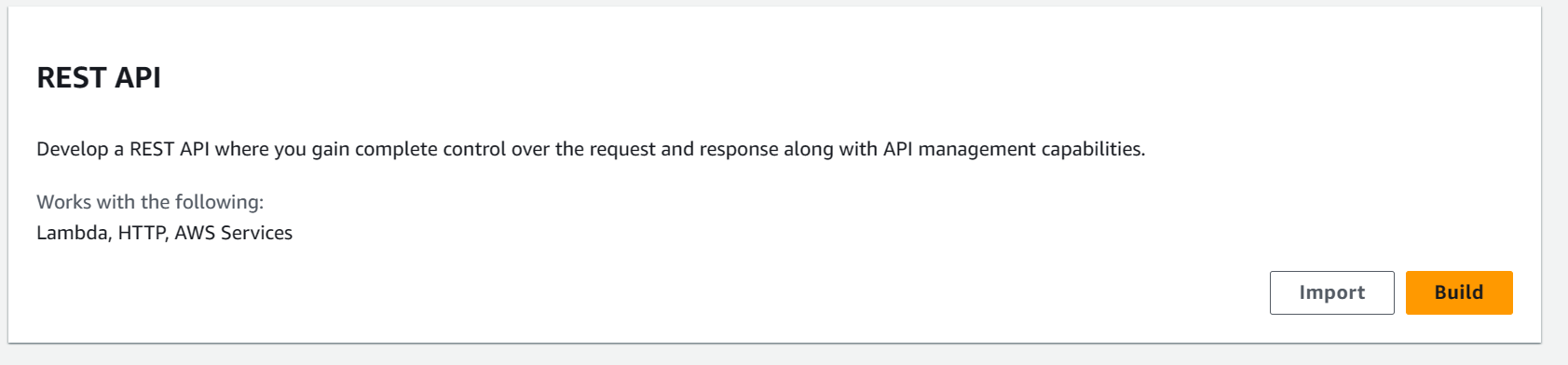
YOU CANE CHECK IT

1. GO DYNAMODB TABLE
2. EXPLORE TABLE
3. CHECK DATA WHICH HAD BEEN GET AND SAVED
4. IN ITEM RETURNED



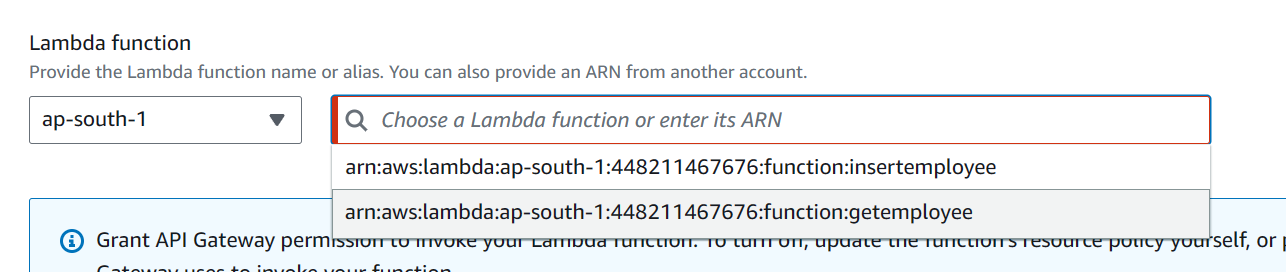
AFTER ALL PROCESS FINAL STEP

CREATE A API GATEWAY

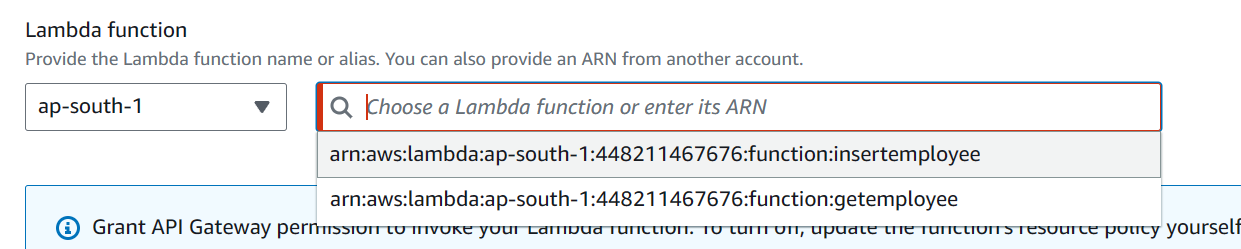


1. Click on BUILD
2. NEW API
3. API NAME – SERVERLESSAPP
4. API endpoint type – Edge-Optimized
5. Create api

CLICK ON CREATE METHOD

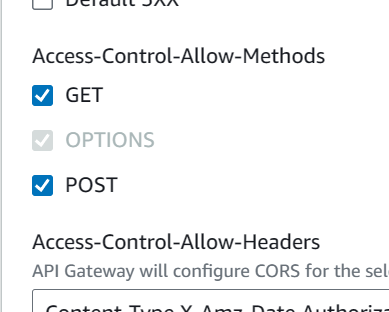
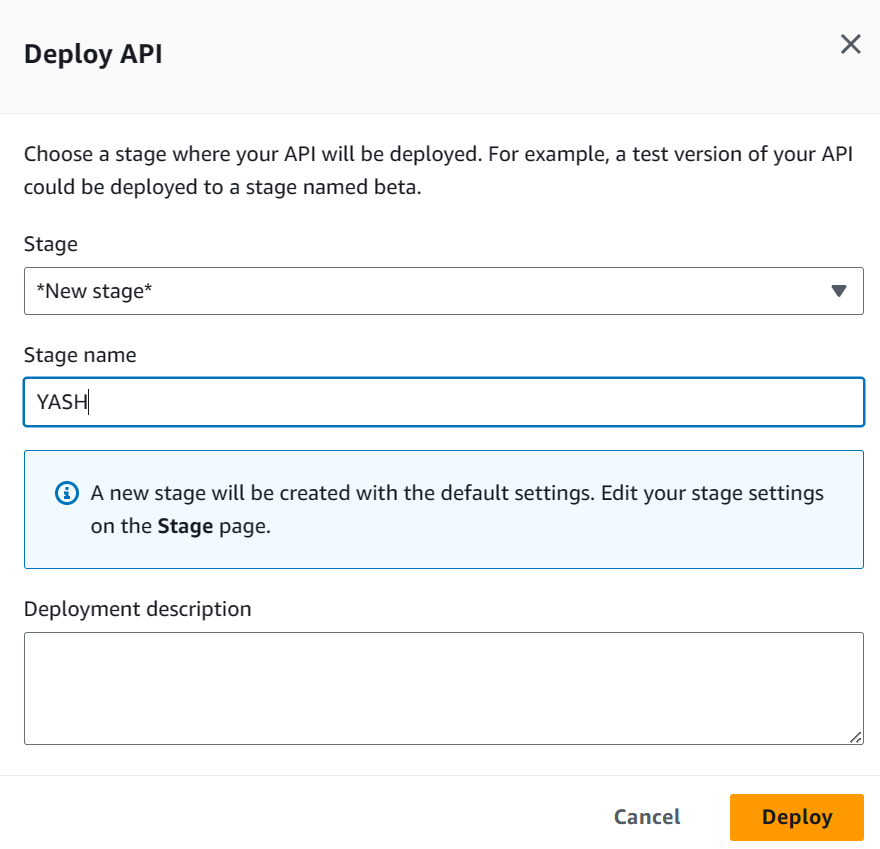
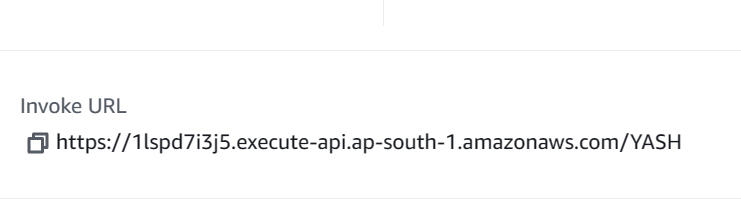
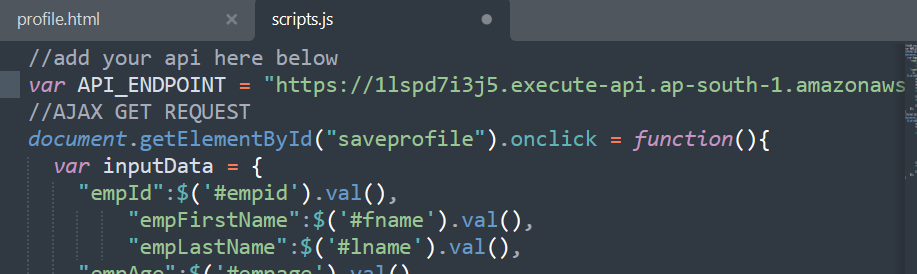
1. METHOD TYPE – GET
2. INTEGERATION TYPE – LAMBDA FUNCTION
3. CHOOSE THE getemployee lambda function
4. 
5. Click on CREATE METHOD

AGAIN WE CREATE METHOD

1. METHOD TYPE – POST
2. INTEGERATION TYPE – LAMBDA FUNCTION
3. CHOOSE THE insertemployee lambda function
4. 
5. CREATE METHOD

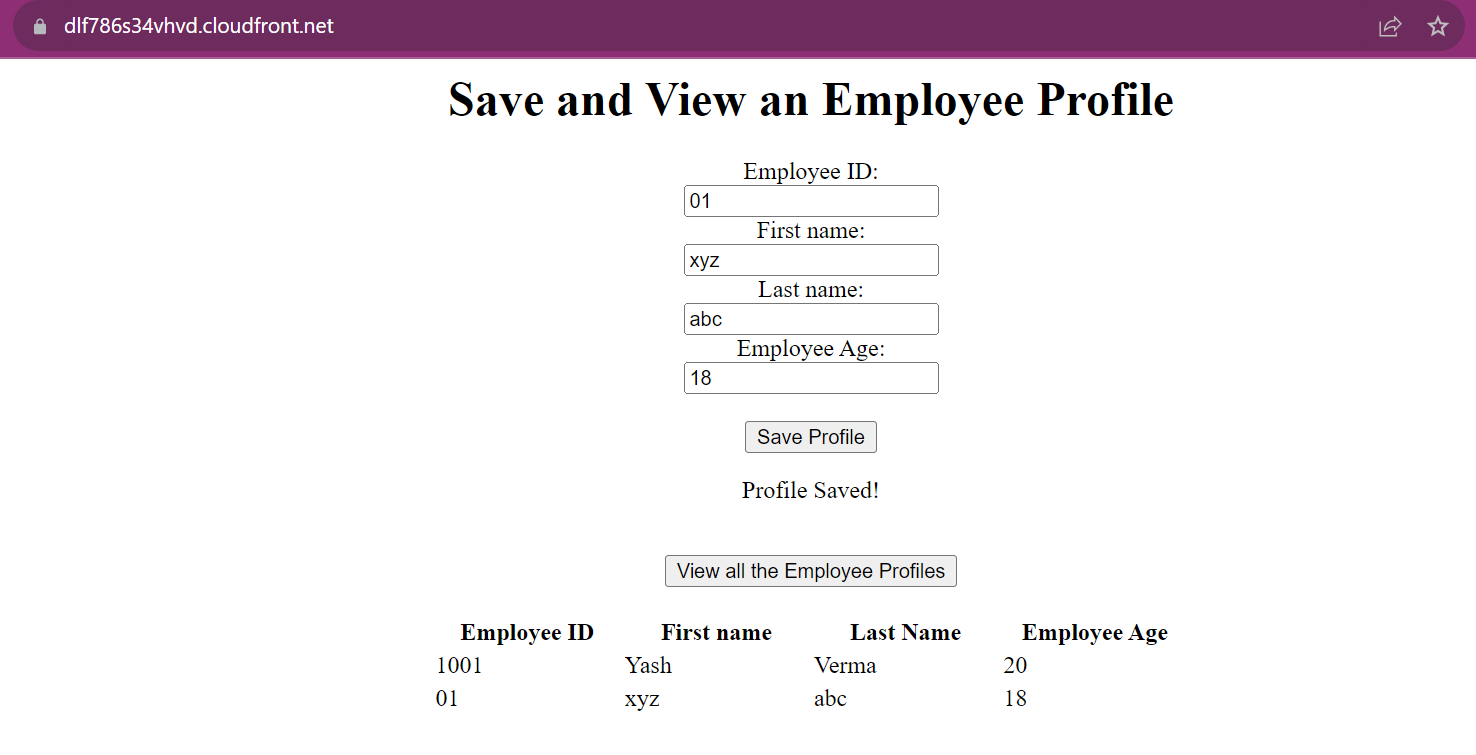
AFTER WE CREATE 2 METHODS

ENABLE CORS

1. CLICK ON ENABLE CORS
2. 
3. TICK MARK BOTH GET AND POST
4. SAVE
5. After save
6. CLICK ON DEPLOY API
7. 
8. CLICK ON DEPLOY
9. NOW WE GET A INVOKE URL
10. 
11. COPY URL
12. PASTE ON script.js file which we created and uploaded on s3bucket
13. Paste it here on code
14. 
15. Save the code
16. Upload it again on s3 bucket

Now copy again distribution domain name link from CLOUDFRONT

Paste link on browser and hit enter



Finally completed our SERVERLESS WEB APPLICATION