ShunyEka System Private LTD – Assignment

Assignment –

1. Deploy a Serverless Static Application using S3, API-Gateway and Lambda Function which ask for user name and return "Hello <User Name>".
2. Using GUI, When User click on a button, API Gateway Trigger Lambda Function and Lambda Function call another AWS Service using "boto3".

Links –

GitHub ->
https://github.com/nirdeshkumar02/DevOps_Repository/tree/master/9.%20Assignment/ShunyEka_Systems_Private_LTD/2.%20Serverless_Html_With_User_Input

Static Website ->

http://contactnirdesh.s3-website.ap-south-1.amazonaws.com

Description – Create a HTML "Contact Us" page which ask for User Data like (Name, Email, and Message), when user click on submit button, An API-Gateway Triggered that route HTTP request to Lambda functions, then Lambda Function called another AWS service "Dynamo DB" where we store user data in database and Return a message "Hello <User Name>! Your data successfully recorded. We will contact you soon."

Steps –

1. **Create HTML Page** –

   Create an 'index.html'. In index.html, create input table for user data (name, email and message) and a submit button also, write javascript function which handle user data and send data through API Gateway.

   Create an 'error.html' which show 'An Error has occurred. Try again later'.

2. **<u>Create an IAM Role</u>** –

Search for IAM -> Go To Roles -> Create Role -> choose a use case 'Lambda' -> click Next -> now, choose '**AmazonDynamoDBFullAccess**' and '**AmazonAPIGatewayAdministrator**' from policies -> Next -> Next -> In Review Section, Provide Role Name ('contact_lambda@role') -> click on Create Role.

3. **<u>Create a Dynamo DB Table</u>** –

Search for Dynamo DB -> Create Table -> Provide Table Name ('contactus') and Primary Key ('email') -> now, click on Create.

4. **<u>Create an API Gateway</u>** –

Search for API Gateway -> Choose 'REST API' and click on Build -> Select Protocol - REST -> Select New API -> Provide API Name ('contact_API') -> Create API.

In Resourse, Select 'create resource' from 'Actions' -> Provide Resource Name ('contactus') and check out to Enable API Gateway CORS -> Create Resource.

Select 'create method' from 'Actions', choose 'POST' and click on 'tick-mark' -> Provide Integration type – Lambda Function, Lambda Region ('ap-south-1') and Lambda Function Name ('contact_function') -> click on Save.

Now, Select 'Enable CORS' from 'Actions' -> click on 'Enable CORS and replace existing CORS headers'.

Select 'Deploy API' from 'Actions' -> Provide Deployment Stage – New Stage, Stage Name – Dev -> click on Deploy -> and now, expand 'Dev' -> select POST -> copy the 'URL' -> and Paste it to HTML Contact Us Page.

5. **<u>Create an Lambda Function</u>** –

Search for Lambda -> Create Function -> Choose 'Author from scratch' -> Provide Function Name ('contact_function'), Runtime ('Python 3.9') -> From Permission, Choose 'use an existing role' and from drop down, choose ('contact_lambda@role') role -> click on Create Function.

In Add Trigger, you will see ‘API Gateway’ is triggered -> Now In Function Code, Write Your Lambda Function (which will accept user input, stored it to database using boto3 and return a message) -> click on Deploy.

6. Create a S3 Bucket –

Search for S3 -> Create Bucket -> Provide Bucket Name (‘contactnirdesh’), Region (‘ap-south-1’) -> uncheck the ‘Block all public access’ -> check the ‘acknowledgement of block all public access’ -> click on Create Bucket.

Select and open Bucket -> click on upload -> select html file (index.html and error.html) -> In Manage Public Permission, Select ‘Grant Public Read Access to this objects’ -> click on Upload.

Now Go to Properties -> Edit the ‘Static Website Hosting’ – Enable -> Provide Index Document – index.html, Error Document – error.html -> click on Save.

Pictures –

a. Lambda Function –

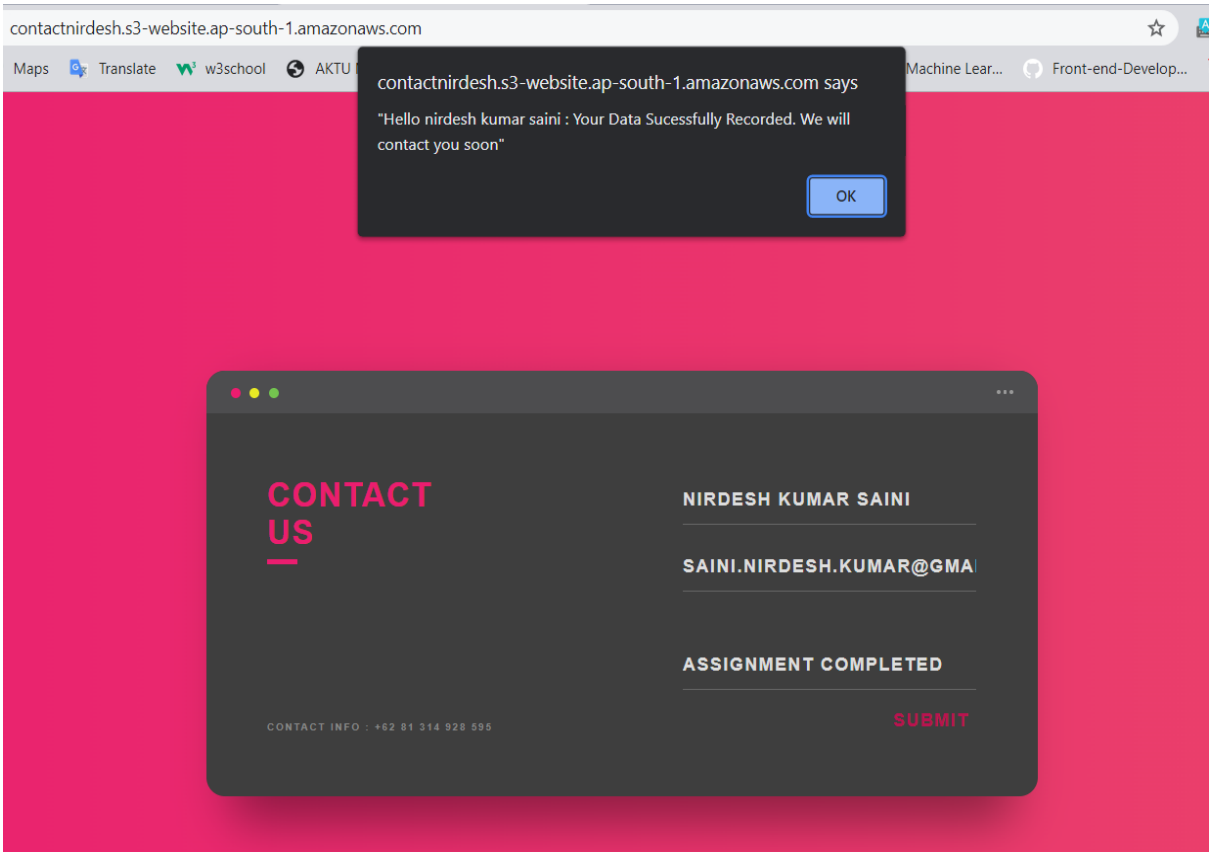
lambda_function

```python
import json
import boto3

dynamodb=boto3.resource('dynamodb')
table=dynamodb.Table('contactus')

def lambda_handler(event, context):
    print(event)
    name=event['name']
    email=event['email']
    message=event['message']
    table.put_item(
        Item={
            'name':name,
            'email':email,
            'message':message,
        }
    )
    submit = 'Hello {} : Your Data Sucessfully Recorded. We will contact you soon'.format(name)
    return submit
```

b. Contact Us



c. DynamoDB