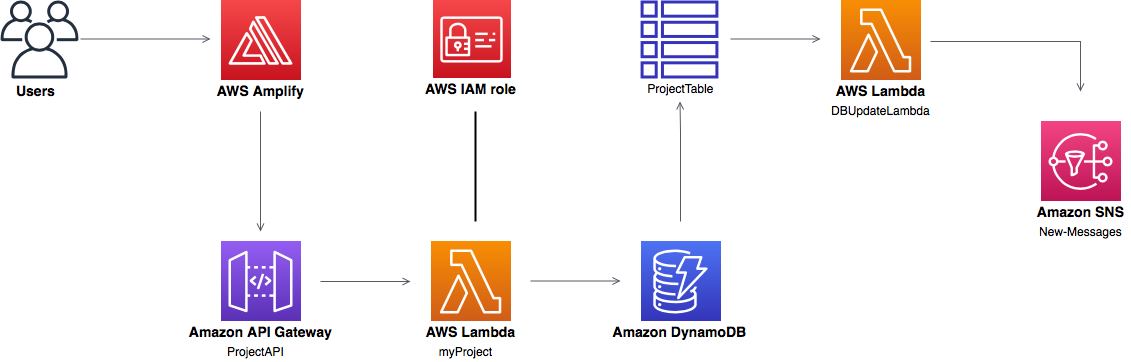
<https://dev.d1edlvkx3u0u87.amplifyapp.com> 🡨 web page URL

In this project, a simple website application is built using AWS Amplify. AWS Amplify stores and deploys static contents like HTML, CSS and JavaScript of the web application. AWS Amplify also exposes the URL which is accessed by end users. Then, the web application is connected with a database via AWS DynamoDB, AWS Lambda and AWS API Gateway for storing the data which comes from a small activity on the webpage – filling out a HTML ‘contact’ form. Configuration of AWS Lambda includes a small piece of code in Python which helps in adding interactivity to the web page. This is piece of code creates a DynamoDB object to select the storage table, extract values from Lambda service and stores in a variable by writing the contents to the DynamoDB table. A RESTful API is created by using Amazon API Gateway to make calls to Lambda function from the web page. A table is created to persists data using Amazon DynamoDB. Then another AWS Lambda function is created to process a stream (when new item is added; updated) from the table created above and then notify the subscribed subscriber via AWS SNS.



Credits:

<https://www.linkedin.com/pulse/aws-lambda-dynamodb-sns-working-together-hamza-tariq>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/US_SetupSNS.html>

<https://aws.amazon.com/getting-started/hands-on/send-fanout-event-notifications/>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.html>

<https://www.styleshout.com/free-templates/ceevee/>

<https://www.antstack.io/blog/create-zip-using-lambda-with-files-streamed-from-s3/>

<https://docs.aws.amazon.com/lambda/latest/dg/getting-started-create-function.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-create-api.html>