**1. Record your reasons for implementing the solution the way you did, struggles you faced and problems you overcame.**

First I gone through the coding assignment that is attached in email, understood the questions clearly & started implementing the solution following the below steps.

1. Created AWS Lambda function in AWS Console with basic IAM role permissions & written the code in my local using Visual studio as IDE with the required node modules & dependencies. Finally zipped the package & uploaded the build in console.
2. Added this lambda function in contact flow in order to use in contact flow.
3. Created a contact flow that greets the customer initially & will invoke the above lambda to find the 3 vanity possibilities based on the phone number & plays back the possibilities back to the customer.
4. In order to play the prompt that says the 3 vanity possibilities more naturally I used speak tags for the message to be played which I have utilized the Amazon Polly service.
5. Once I done with the creation of backend lambda, contact flow & its integration in the contact flow I started creating the API that fetches the vanity possibilities as a response based on the customer phone number as a path param using Amazon APIGateway.
6. For getting the vanity possibilities based on phone number created one more lambda that will be called by the Amazon API Gateway.
7. After that I deployed the API by enabling CORS & copied the API Gateway endpoint.
8. Created a small webapp application using app.js file where in I called the above API gateway point to fetch the response using axios & to show in the frontend.
9. Once I done with the webapp, finally I started writing the cloudformation template according to the use case requirement stated in the email.

**b. What shortcuts did you take that would be a bad practice in production?**

According to my point of view, we need to make sure that every resource whatever we are going to define in our cloudformation template must have unique name so that resource creation will never get failed if we want to deploy the same template/creation of resources in multiple regions.

Before to start testing the lambda’s we need to make sure that correct IAM permissions has been given in order to save the time instead of adding one by one permissions after we receive the errors in CloudWatch logs.

**c. What would you have done with more time?**

If I had more time, I mostly utilize to learn the skills/ the ones which I don’t know that helps in my career growth in future. I always looking forward to learn more.

**d. What other considerations would you make before making our toy app into something that would be ready for high volumes of traffic, potential attacks from bad folks, etc.**

First, we need to make sure our application architecture is scalable & can handle increased loads by load balancing. Before our app into something its very important to perform performance testing whether the app remains stable & highly responsive under high traffic conditions. Better to improve security including firewalls & encryption.

Before to start testing the lambda’s we need to make sure that correct IAM permissions has been given in order to save the time instead of adding one by one permissions after we receive the errors in CloudWatch logs.

**e. Please include an architecture diagram.**

A diagram of a customer service

Description automatically generated