



Project Report

Fall – 2020

LAB: REAL-TIME VOICE TRANSLATION APPLICATION

Aarya Shekhaar Tilekar

U36766567

Guided by Dr Phil Ventura, Professor, University of South Florida

Lab: Real-Time Voice Translation Application

INTRODUCTION

Just imagine - you say something in one language, and a tool immediately translates it to another language. Wouldn't it be even cooler to build your own real-time voice translator application using AWS services? It would be similar to the Babel fish in *The Hitchhiker's Guide to the Galaxy* [1]:

"The Babel fish is small, yellow, leech-like—and probably the oddest thing in the universe... If you stick one in your ear, you can instantly understand anything said to you in any form of language."

Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

In this post, we will see how you to connect multiple services in AWS to build your own application that works like a bit like the Babel fish [1].

LAB OBJECTIVE

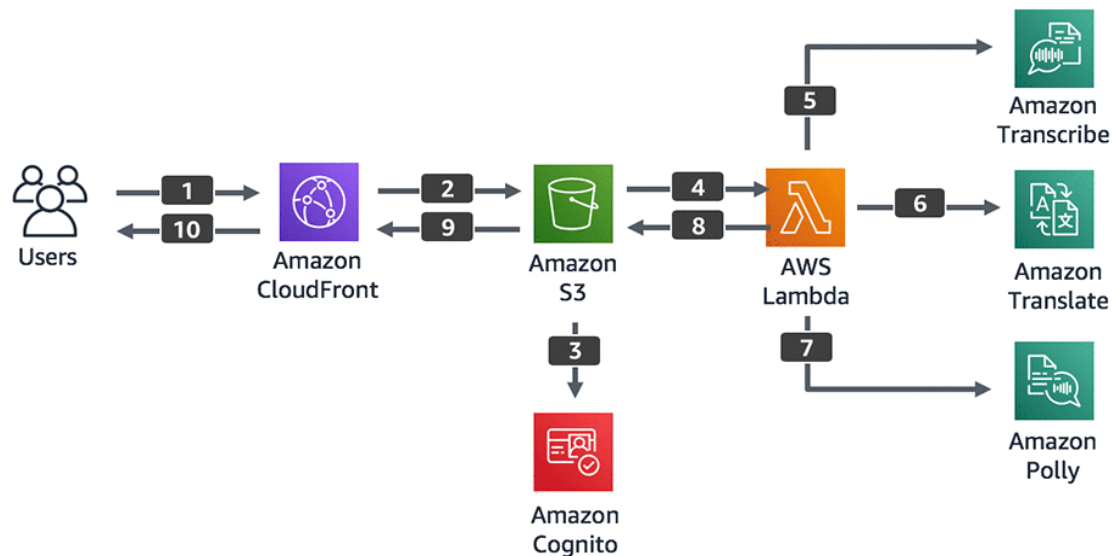
The heart of this application consists of an [AWS Lambda](#) function that connects the following three AI language services:

- [Amazon Transcribe](#) - This fully managed and continuously trained automatic speech recognition (ASR) service takes in audio and automatically generates accurate transcripts. Amazon Transcribe supports [real-time transcriptions](#), which help achieve near real-time conversion.
- [Amazon Translate](#) - This neural machine-translation service delivers fast, high-quality, and affordable language translation.
- [Amazon Polly](#) - This text-to-speech service uses advanced deep learning technologies to synthesize speech that sounds like a human voice [1].

A diagrammatic representation of how these three services relate is shown in the following illustration [1].



To make this process a bit easier, we will use an [AWS CloudFormation](#) template, which initiates the application. The following diagram shows all the components of this process, which is later described in detail [1].



In this tutorial, you will perform following tasks:

1. Allow access to your site with [Amazon CloudFront](#), which allows you to get an HTTPS link to your page and which is required by some browsers to record audio.
2. Host your page on [Amazon S3](#), which simplifies the whole solution. This is also the place to save the input audio file recorded in the browser.
3. Gain secure access to S3 and Lambda from the browser with [Amazon Cognito](#).
4. Save the input audio file on S3 and invoke a Lambda function. In the input of the function, provide the name of audio file (that you saved earlier in Amazon S3), and pass the source and target language parameters.
5. Convert audio into text with Amazon Transcribe.
6. Translate the transcribed text from one language to another with Amazon Translate.
7. Convert the new translated text into speech with Amazon Polly.
8. Save the output audio file back to S3 with the Lambda function, and then return the file name to your page (JavaScript invocation). You could return the audio file itself, but for simplicity, save it on S3 and just return its name.
9. Automatically play the translated audio to the user.
10. Accelerate the speed of delivering the file with CloudFront [1].

STEPS

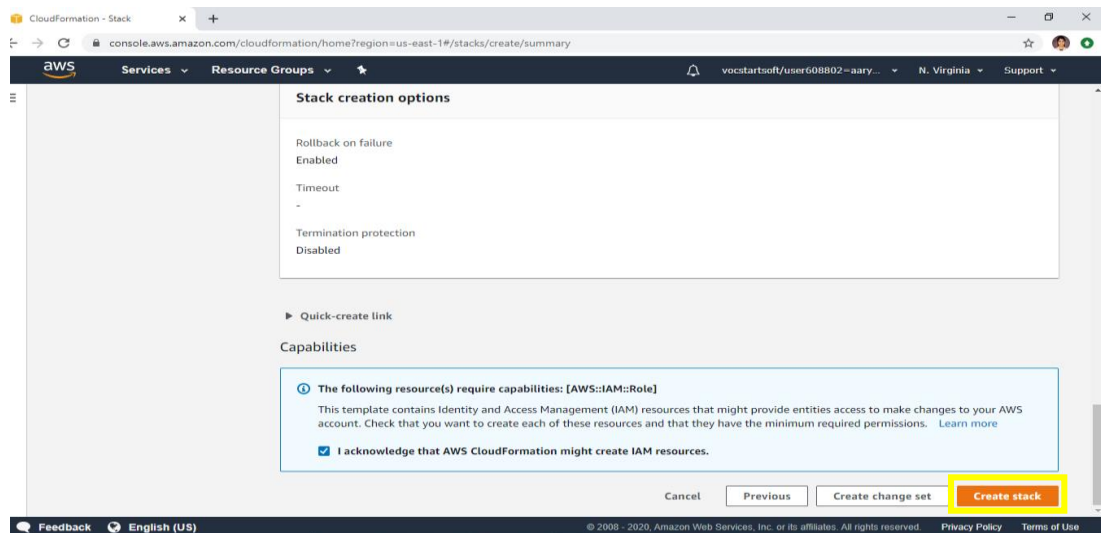
As mentioned earlier, we created an AWS CloudFormation template to create all the necessary resources.

I. LAUNCHING STACK

Sign into the console, and then click on [LaunchStack](#), which launches a CloudFormation stack in your AWS account. The stack launches in the US-East-1 (N. Virginia) Region [1].

II. CREATING STACK

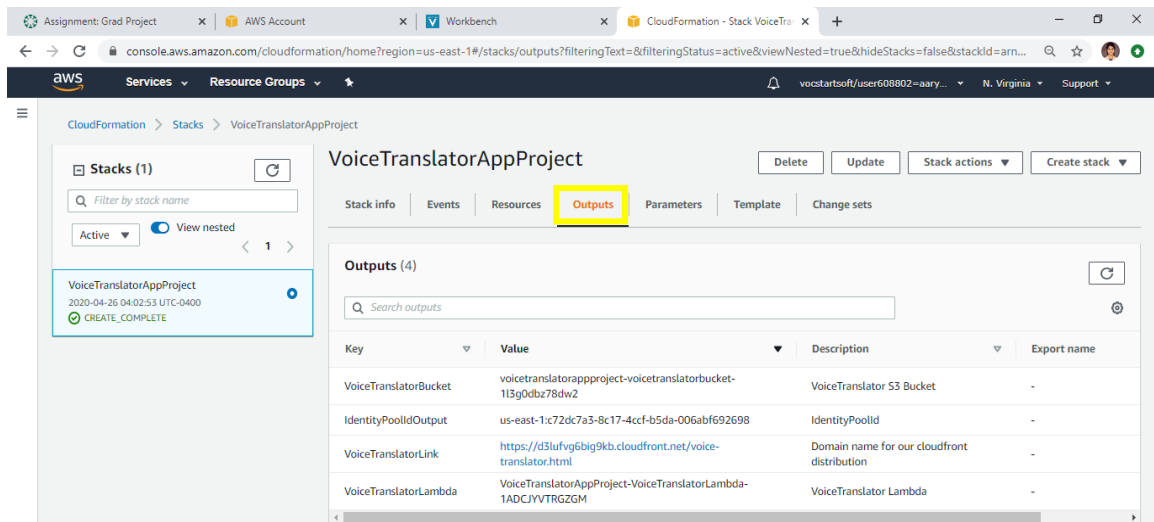
Go through the wizard and create the stack by accepting the default values except for the name of the stack which you can change as per your desire; I have renamed the stack as `VoiceTranslatorAppProject` myself. On the last step of the wizard, acknowledge that CloudFormation creates IAM After 10 - 15 minutes, the stack has been created [1].



III. OUTPUTS SECTION OF STACK

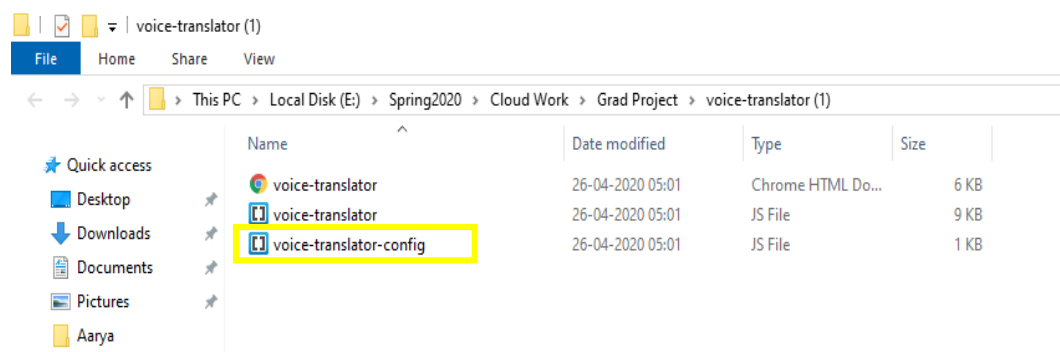
In the Outputs section of the stack shown in the following screenshot, you find the following four parameters:

- *VoiceTranslatorLink*—The link to your webpage.
- *VoiceTranslatorLambda*—The name of the Lambda function to be invoked from your web application.
- *VoiceTranslatorBucket*—The S3 bucket where you host your application, and where audio files are stored.
- *IdentityPoolIdOutput*—The identity pool ID, which allows you to securely connect to S3 and Lambda [1].



IV. FILES DOWNLOAD

Click on this [link](#) to download the zip file and then unzip it. There are three files included [1].



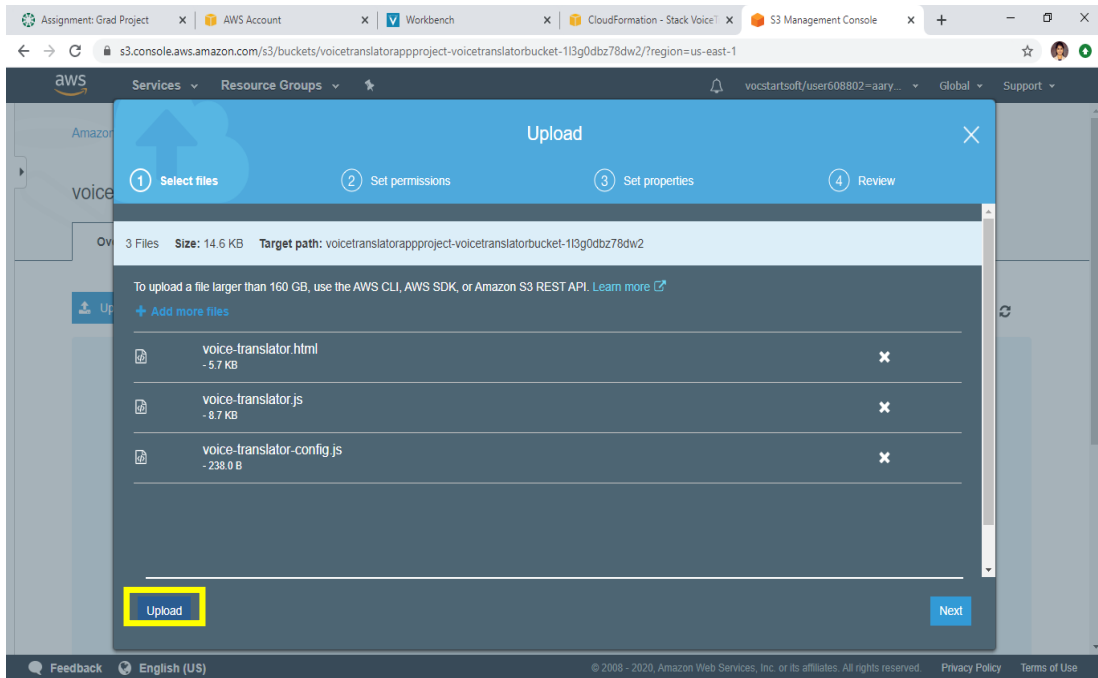
V. EDIT voice-translator-config.js

Open the downloaded file named *voice-translator-config.js*, and edit it based on the four output values in your stack (Step 3). It should then look similar to this [1].

```
var bucketName = 'voicetranslatorapp-voicetranslat.....';
var IdentityPoolId = 'us-east-1:535.....';
var lambdaFunction = 'VoiceTranslatorApp-VoiceTranslatorLambda-....';
```

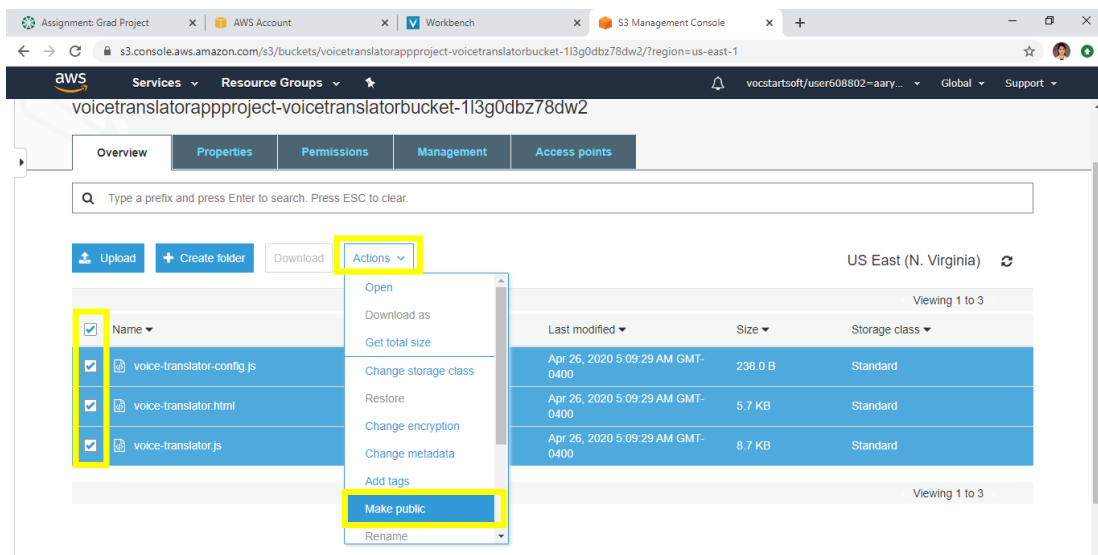
VI. S3 BUCKET

In the S3 console, open the S3 bucket (created by the CloudFormation template). Upload all three files, including the modified version of *voice-translator-config.js* [1].



VII. MAKE PUBLIC

After uploading the files, select the three files and make them public by clicking the **Action** button and selecting **Make Public**.

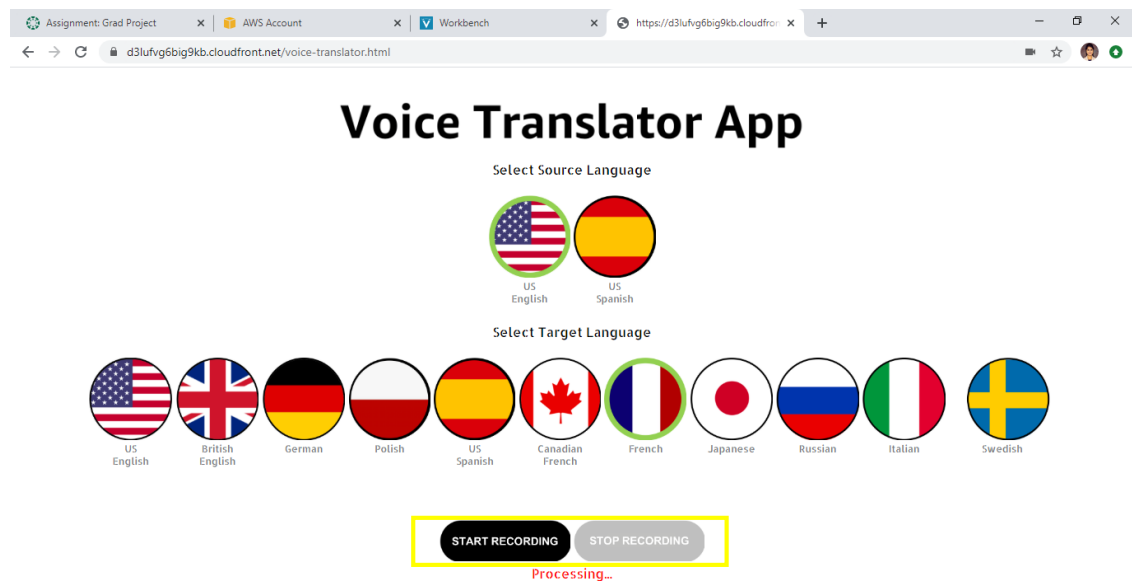


VIII. TESTING

Open your application from the link provided in Step 3. In the Voice Translator App interface, perform the following steps to test the process:

- Choose a source language.
- Choose a target language.
- Think of something to say, choose **START RECORDING**, and start speaking.
- When you finish speaking, choose **STOP RECORDING** and wait a couple of seconds.

If everything worked fine, the application should automatically play the audio in target language [1].



IX. CONCLUSION

As you can see, it takes less than an hour to create your own unique voice translation application, based on the existing, integrated AI language services in AWS. Plus, the whole process is done without a server.

This application currently supports two input languages: US English and US Spanish. However, Amazon Transcribe recently started supporting real-time speech-to-text in British English, French, and Canadian French. Feel free to try to extend your application by using those languages [1].

X. OUTPUT

Link for this application - [Voice Translator App](https://d3lufvg6big9kb.cloudfront.net/voice-translator.html)

REFERENCES:

1. <https://aws.amazon.com/blogs/machine-learning/build-your-own-real-time-voice-translator-application-with-aws-services/>