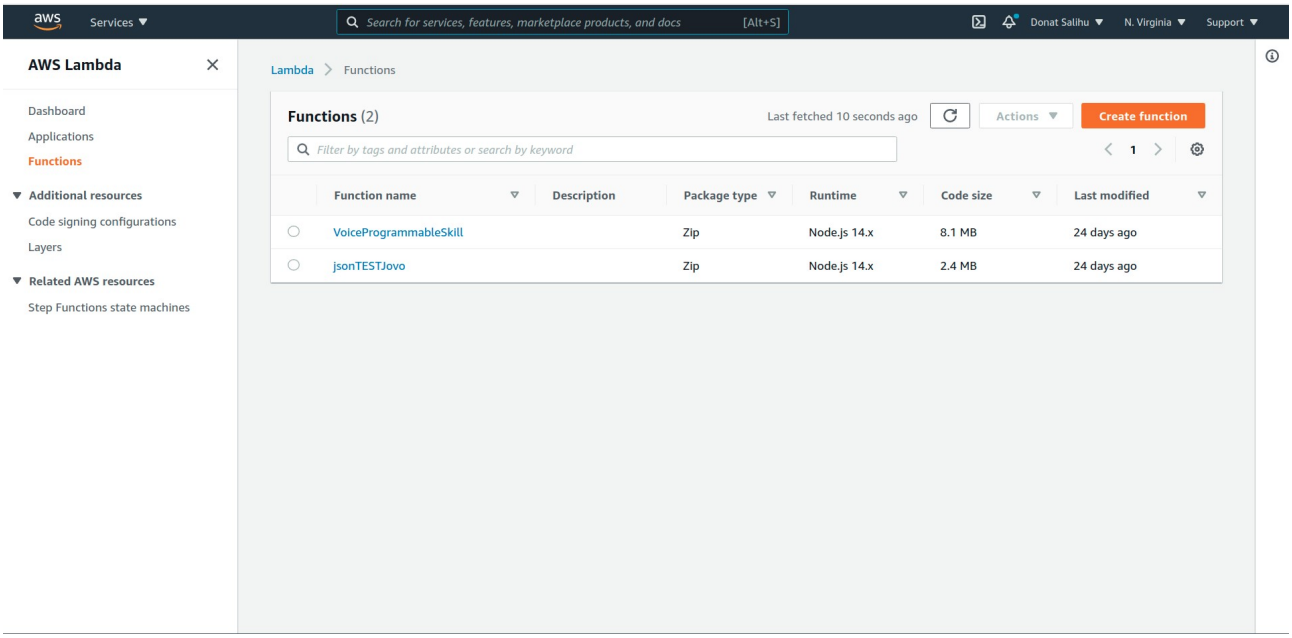
Installation Manual

This manual assumes you have an Alexa Developer Console account and AWS account.

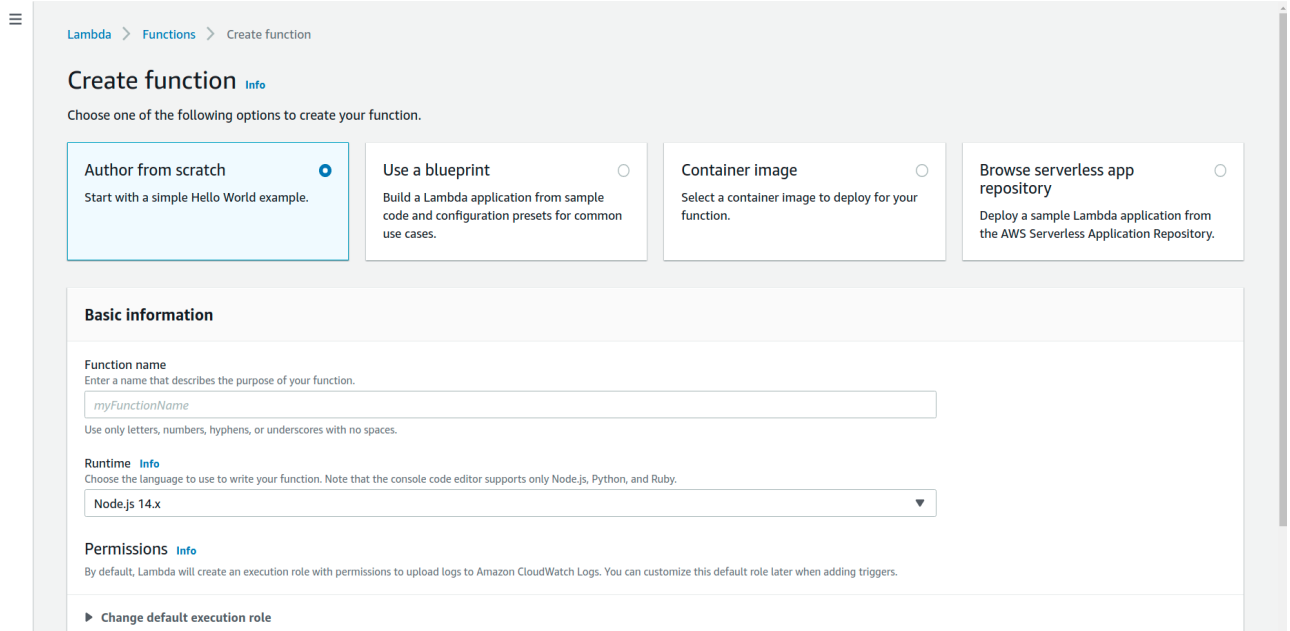
There are two files containing two different skills. The first skill is the “JovoSelfProgrammable” skill. In order to deploy the skill on the Alexa Developer Console for testing purposes, the Jovo framework should be installed on the local machine. To install the Jovo CLI open the terminal and enter: “$ npm install -g jovo-cli”. The next step is to install the CMS for Google Sheet integration:

$ npm install --save jovo-cms-googlesheets. To deploy the project: open the terminal path to the project and run: “$ jovo build”, this command is going to build the model for an Alexa skill. The next step is to deploy the skill using: “$ jovo deploy --platform alexaSkill” Now that the skill has been deployed to the Alexa Developer Console, the code should be hosted on the Lambda Function as well. A way to do that is to go to the AWS console and navigate to the Lambda Function. Create a new function from scratch. After the funciton has been created, go to the terminal and run “$ npm run bundle”, this command is going to bundle the src folder needed for the Lambda Function.

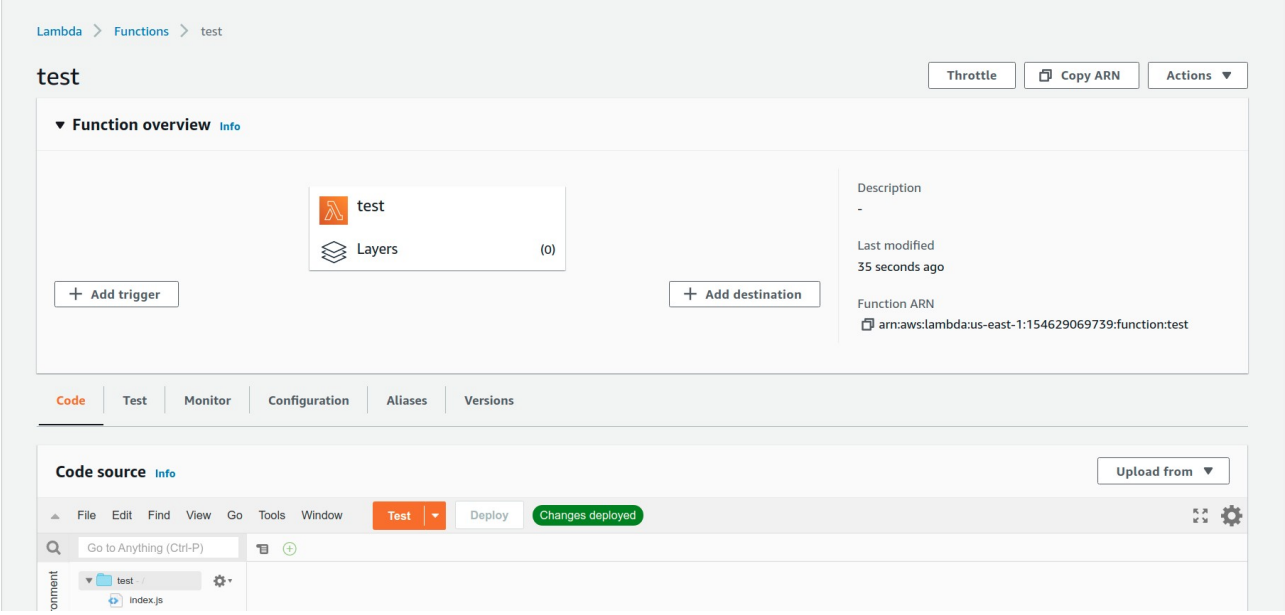
At the newly created Lambda Function from scratch go to the +add trigger option and add Alexa Skill Kit. The next step is to deploy the bundle folder created earlier. Go to the upload option on the Lambda Function and upload from your local machine the bundle file created when “$ npm run bundle” was run.



The Create function option to create a new Lambda Function to host the skill code



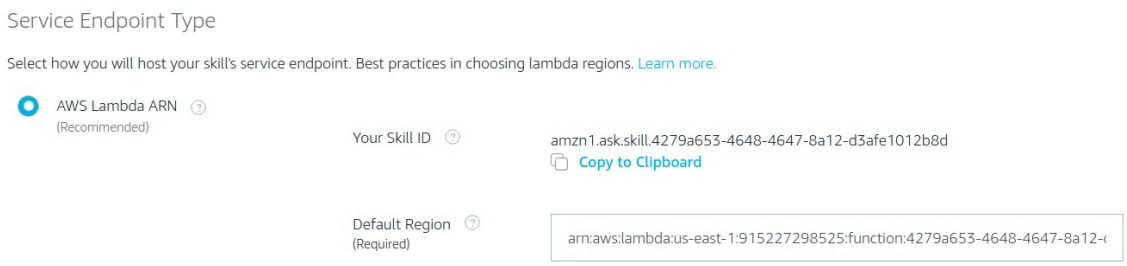
Creating a skill from the scratch, name the skill and press the create function option.



The +Add trigger button to add the Alexa Skill Kit tirgger and the Upload from option to upload the bundle folder created form the skill.

After the skill has been deployed to the Alexa Developer Console and the the bundle file in Lmaba Function, the next step is to copy the ARN of the Lambda Function which can be found just next to

the +add trigger and +add Destination, look at the above picture. After the Arn has ben copied, to the Alexa Developer Console at the Jovo skill you just deployed, navigate to the ENDPOINT option and paste the arn code.

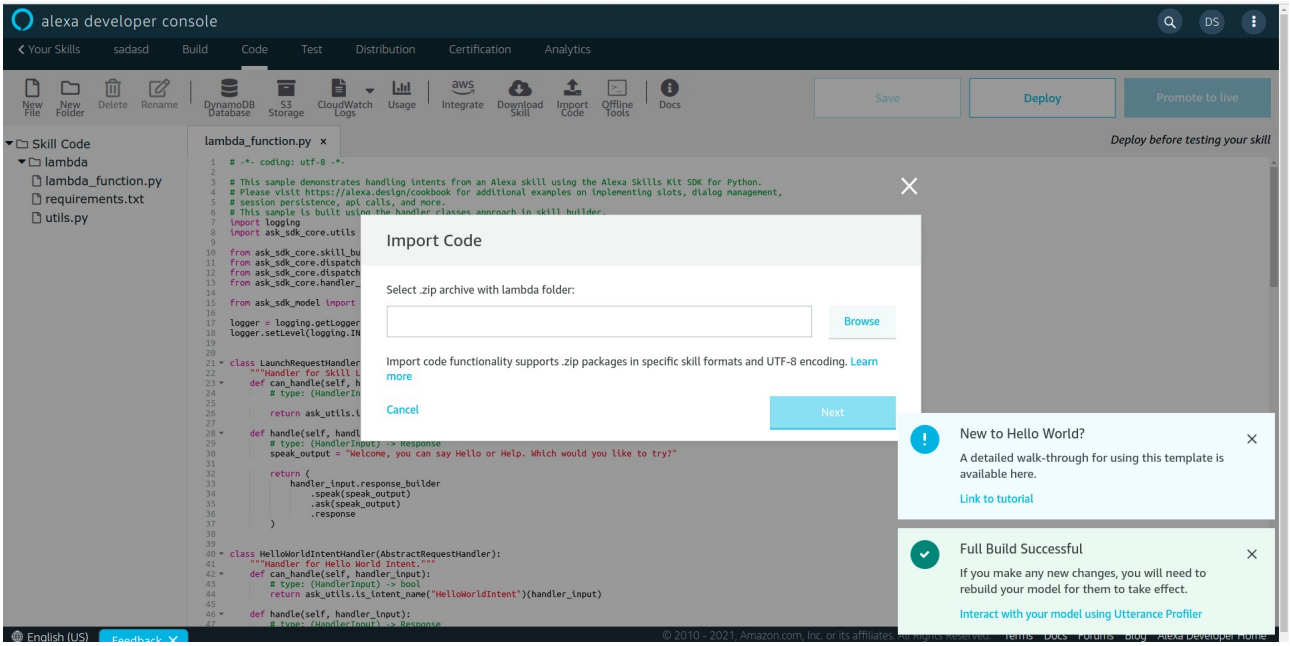


Click the AWS Lambda ARN and at the Deafult region paste the Lambda Function ARN code.

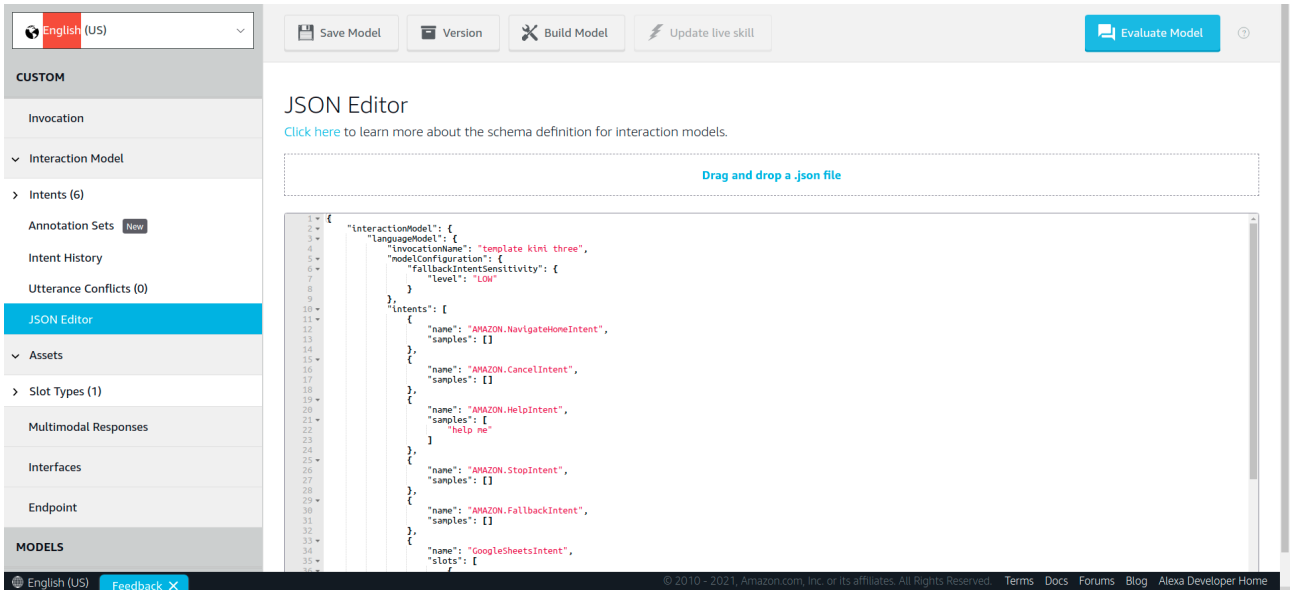
Press build on the console and the deployment of the first Skill is finished.

For the second skill, go to Alexa Developer Console, and create a new skill.

Enter the skill name, the model of the skill should be CUSTOM, and the method to host the skill backend resources should be Alexa-Hosted(Python). And when asked to the template, select start from scratch. After the skill was created, go to the code option on the console, and press the upload code option. Import the the file contining the second skill and then press deploy.



After the code is uploaded and deployed, go to the build option, navigate to the interaction model and open the JSON Editor. There is an option drag and drop a .json file. Click that option and upload the VoiceProgrammingSkill/interactionModels/custom/en-US.json.



The second skill is ready as well.

///

The url to the google sheet is: https://docs.google.com/spreadsheets/d/15ZsXiUPpKHdZeLCNAoBdfGvVJCED7M9b0494KfALS0o/edit#gid=0

I made the link public so anyone testing the system would not have to create a new Sheet and go to the steps of getting APIs and credentials but instead you the Sheet I used which is linked with both of the skills though APIs.