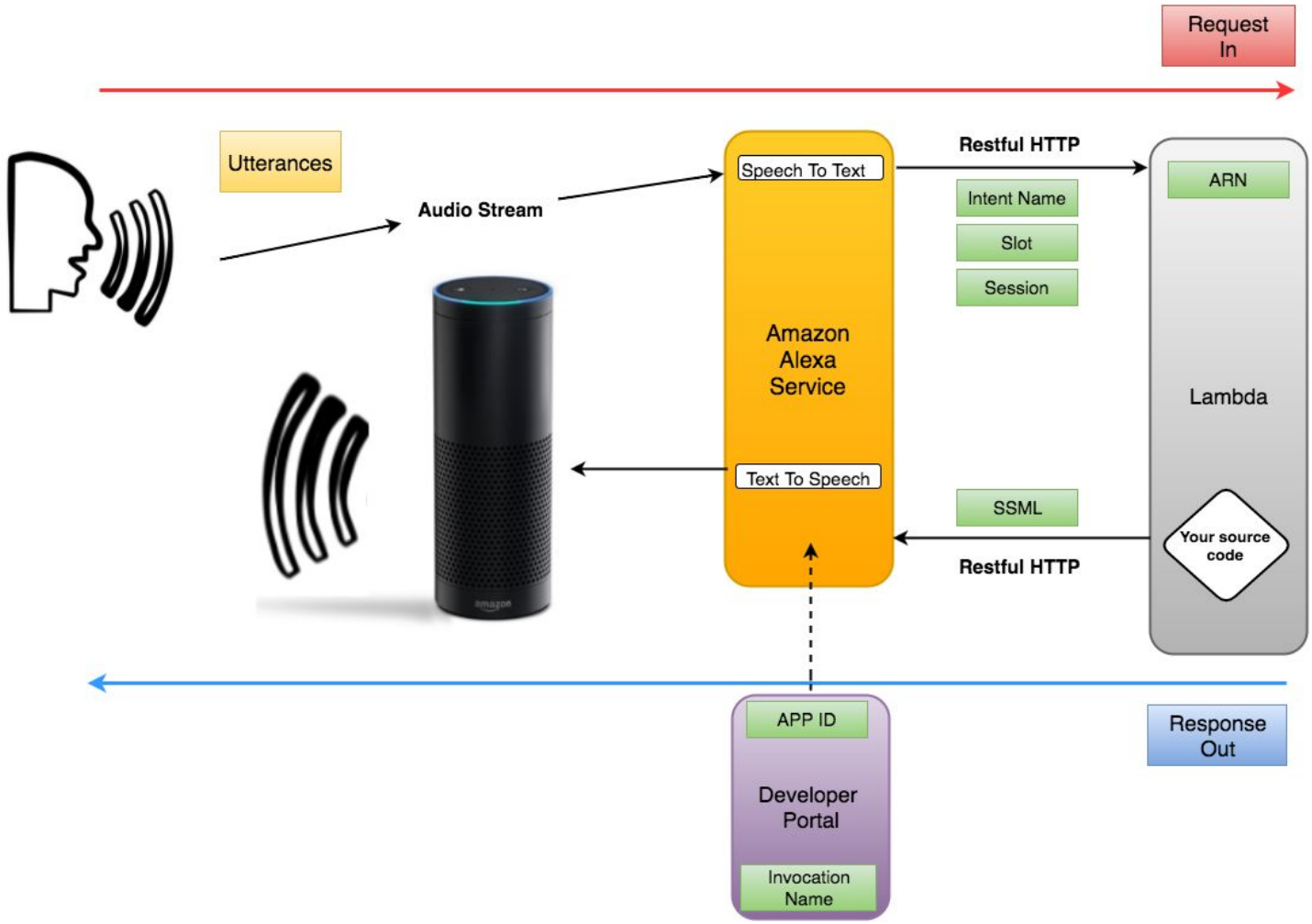


Glossary

Alexa/ Alexa service The cloud-based service that Echo uses which is capable of listening and responding to commands. It is being used in different devices. e.g Echo, Tap, smart watches and even car.
Echo A smart home device that is connected to the cloud and uses Alexa.
Intent An action that is invoked when a user utters a command or a statement
Slot A variable that is captured from the utterance and is given to the intent
Intent Schema A JSON file that describes the names and slots of all intents available within a skill.
Sample Utterance Example phrases that help train Alexa to understand how utterances map to an intent. This is not an enumeration/definitive list of phrases that can trigger an intent.
Lambda function A function that runs inside of the Lambda Service and it is triggered in response to events.
ARN (Amazon Resource Name) This name is like a URL which helps the Alexa service to know which resource to call. For instance, a lambda function.
Developer Portal Web interface that allows developers to configure their applications. For instance, the invocation name of the skill, sample utterances, intent schema and the ARN of their lambda functions
Session A place to temporarily save information in between intents.
Handler and Listeners A handler is an object that contains multiple methods called listeners. A listener is a function that gets invoked in response to when an event is emitted.

Project Setup

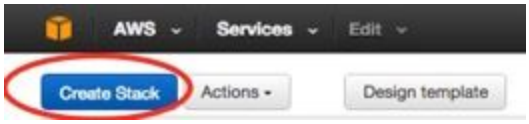
1. **Code checkout** by using the following command lines. **Note** that this code is heavily documented for your benefit
    - git clone https://github.com/mandnyc/Alexa-Hackathon-Quick-Starter.git
    - cd hackathon-quick-start/
    - npm install

2. **Configuration**
    - Sign in **AWS console**

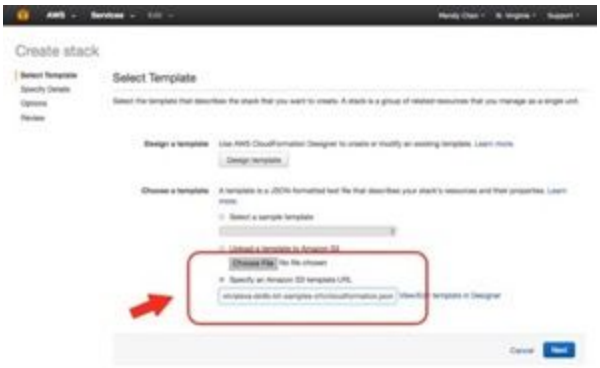
        - Log in to https://console.as.amazon.com
        - In the top navigation, look for **Services**, then look for **Management Tools** under **All AWS Services**, select **CloudFormation**

    - In **CloudFormation**

        - Click the blue button labeled **Create Stack**

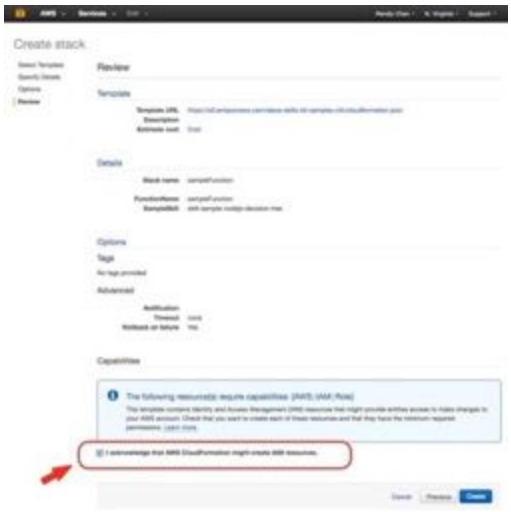


        - Then you will see the screen below calls **Select template**, choose **Specify an Amazon S3 template URL**
        - Paste this S3 URL https://s3.amazonaws.com/alexa-skills-kit-samples-cfn/cloudformation.json



        - In the image below, under **Parameters**, enter in **FunctionName** and select the template under the **SampleSkill** dropdown menu
        - Click **Next**

- In the next screen you'll see the below image. check the box "**I acknowledge that AWS CloudFormation might create IAM resources**"

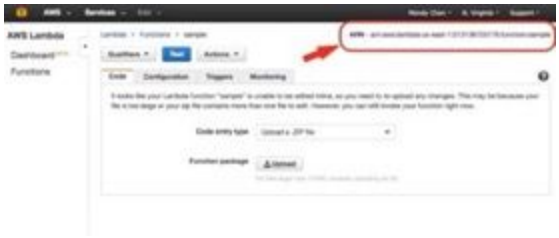


- While the CloudFormation task is creating the Lambda Function, Click on the navigation menu item **Services** again. Locate **Compute** under **All AWS Services**, then select **Lambda**

○ In Lambda

- On this screen, you'll see a table which has a column labeled **Function Name.** Locate the function name you created in the steps above and click on the name.

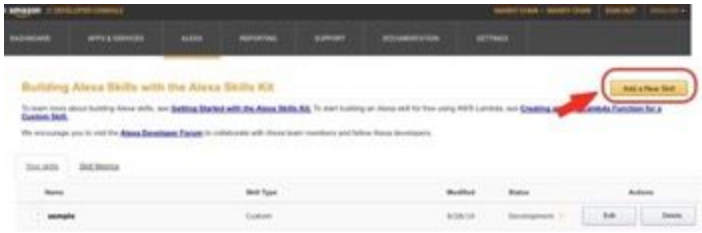
- Locate the **ARN** in the upper right hand corner and save it to a file for later. We will use this in the developer portal at https://developer.amazon.com/ and sign in.



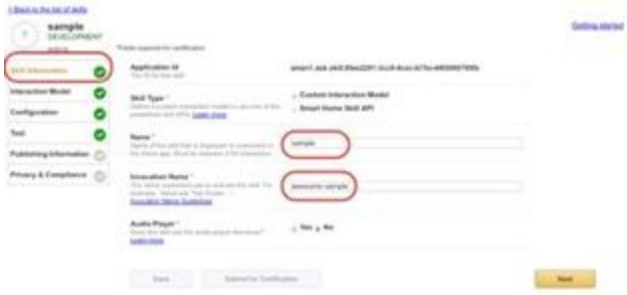
- In the top navigation of the Developer Portal, locate **Alexa** and click on it.

○ In Developer Portal

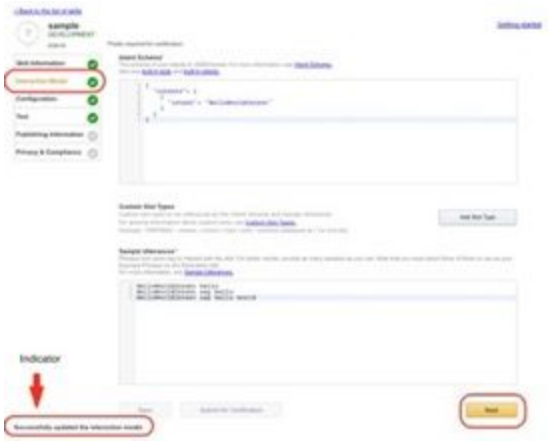
- Select **Alexa Skill Kit**, click **Get Started**
- Click on the yellow button labeled **Add a new skill** on the top right corner



- On the next screen, you'll see a navigation menu on the left. Locate the menu item **Skill information**. This is where you decide the **name** and **innovation name** of the skill.



- Under **Interaction Model**, paste the **intent Schema** and **Sample Utterances** from the SpeechAssets folder of our application **hackathon-quick-start**



- When you click on the yellow button labeled **Next**, you'll notice a little activity indicator in the lower left corner. This means Alexa Service is creating an interaction model for your skill.

- Under **Configuration**, retrieve the **ARN** you saved earlier and paste it into the **Endpoint**. Make sure you click the radio button labeled **Lambda ARN (Amazon Resource Name)**

3. Deploy

- You must first install the Amazon Command Line Interface (CLI) in order to publish your skill to Lambda using the provided publish.sh
- Follow the instruction created by Robert McCauley, Amazon Alexa Solution Architect
- https://developer.amazon.com/public/community/post/Tx1UE9W1NQ0GYII/Publishing-Your-Skill-Code-to-Lambda-via-the-Command-Line-Interface
- Once the CLI is installed, go to your terminal, cd to your project directory and run the **publish.sh** script to deploy the skill.

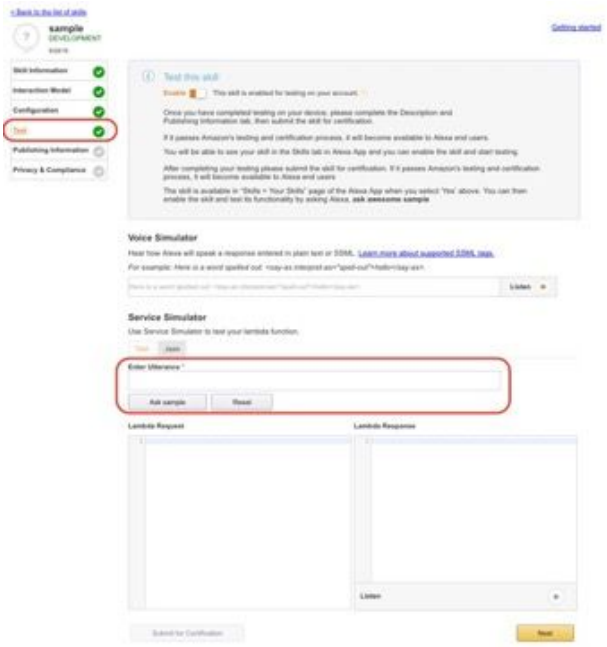
  Reminder : before you run the publish.sh, make sure the function name is updated

```
rm index.zip
cd src
zip -X -r ../index.zip * .env
cd ..
aws lambda update-function-code --function-name <function name> --zip-file fileb://index.zip
```

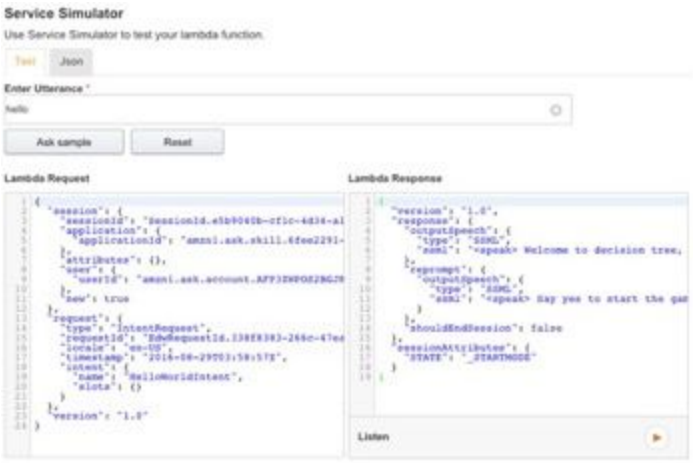
```
mandys-MacBook-Pro-2:hackathon-quick-start mandychan$ cd /Users/mandychan/temp/hackathon-quick-start
mandys-MacBook-Pro-2:hackathon-quick-start mandychan$ ./publish.sh
```

4. Test

- There are two ways to test your code without using the device
  - Developer Portal
  - https://www.Echosim.io
- In the **Developer Portal**, once again click on the Alexa menu item at the top.
- Then click on **Get Started** under the **Alexa Skills Kit** to return to the skill we previously configured
- Click on the name of your skill in the **Your Skills** table.
- On the left, click on the **Test** menu item to bring up the Voice and the Service Simulator as shown below.



- Enter in your test utterance in the **Enter Utterance** input field and click next to see the results.
- A successful test should look like the below image.



- The second method is using https://echosim.io/ - Alexa in the browser where you can speak to your skill.