**ALEXA, WHO IS AT THE DOOR? - AN AMAZON ALEXA AND RASPBERRY PI PROJECT THAT ALLOWS A USER TO KNOW WHO EXACTLY IS KNOCKING AT THEIR DOOR USING FACIAL RECOGNITION.** **\*Taofeek Obafemi-Babatunde.** Morgan State University, Department of Electrical and Computer Engineering, Baltimore, MD 21251.

**Overview:**

An Amazon Alexa and Raspberry Pi project that allows a user to know who exactly is knocking at their door using facial recognition.

**Schematics:**

A screenshot of a cell phone

Description automatically generated

**Things used in this project:**

Hardware:

* Raspberry Pi 2 Model B
* Amazon Echo (Amazon Alexa)
* Raspberry Pi camera module

Software and online services:

* Amazon Alexa skills kit
* Firebase
* Amazon web services (AWS) Lambda

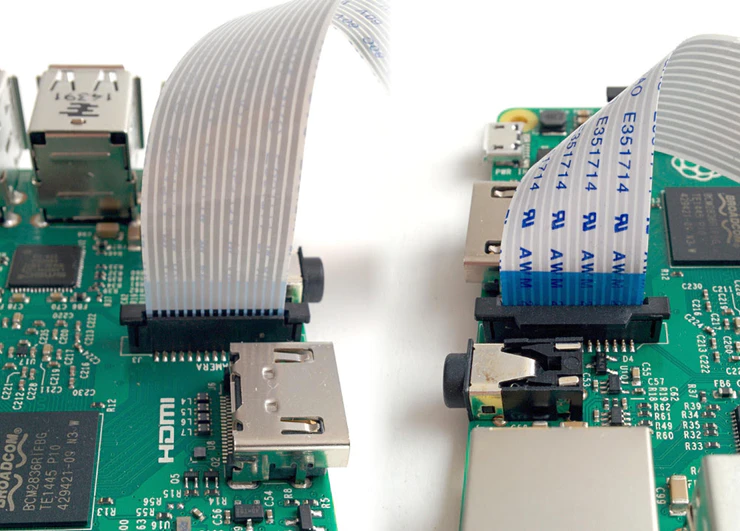
**Introduction:**

“Alexa, who is at the Door?” is an Amazon Alexa skill set that utilizes Raspberry Pi and Firebase along with a facial recognition API to let you know who is knocking at your door. After Alexa gets the approval to take a picture of the visitor’s face, it sends a command to the Raspberry Pi to take a picture of them using the camera module. The picture is then processed through a facial recognition API called “Kairos” which then determines whether the face is recognized, unknown, or not there at all.

After Alexa is taught a new face, the person is given a name by the user which is then saved in the database. It can be re-taught several times in each occasion that the visitor comes to your home. This way, the visitor’s face can be recognized in different conditions such as brightness, darkness, or a slight change in appearance of the individual. This also improves Alexa’s speed at which she recognizes each new friend that is added to the database.

**Setting up the Raspberry PI and Camera Module:**

One of the key components of this project is setting up the Raspberry Pi and the camera module properly because after all, we wouldn’t be able to see who has wandered up to our door! After installing the Raspbian operating system onto your pi from Raspberry Pi’s website, we’re going to need to need to install the latest kernel, GPU firmware and applications for Raspberry Pi. The “RaspiCam” Documentation provided on Raspberry Pi’s website makes this super easy for us.



Credits: raspberrypi.org

* Step 1: Execute the following commands into your Raspberry Pi terminal to get started.

sudo apt-get update

sudo apt-get upgrade

* Step 2: Next, we need to enable RaspiCam camera support.

sudo raspi-config

* Step 3: When the fancy GUI appears, use the arrow keys on your keyboard to move down to enable and hit the enter key to select it. After exiting raspi-config, you will be prompted to reboot your Pi. The enable option ensures that upon reboot the correct camera drivers will be running. To test that everything is working smoothly try the following command in your terminal.

raspistill –v –o test.jpg

If everything is going as planned, the Raspberry Pi display should show a 5-second preview from the camera’s perspective and you should see a little red dot appear in the top corner of the camera module. The terminal will output various informational messages as this is happening. After the picture is taken, it’ll be saved under in the file test.jpg which you can then open and view your marvelous picture!

If things aren’t working as they should be, the RaspiCam Documentation outlines a series of very descriptive troubleshooting methods that will help you get your Pi camera up and running in no time.

**Setting up Database and Configure Code Repository:**

For "Who's at the Door?", firebase was used as the connection between local and cloud communication. Both the lambda function and the Raspberry Pi are listening to the firebase DB (the lambda function only listens to firebase when it's triggered by Alexa Voice Services).

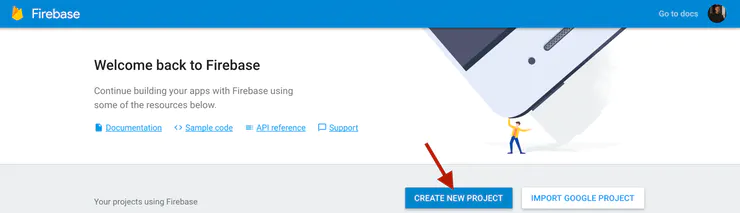


Based on messaging triggers sent by both the Lambda Function and the Raspberry Pi in the flow of this Alexa Skill, specific responses can occur on the Alexa and specific actions can occur on the Raspberry Pi.

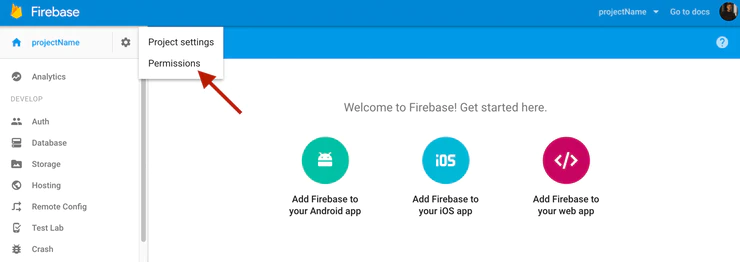
**Setting up Firebase Service:**

Step 1: Create a firebase account or login with your google account at Firebase.

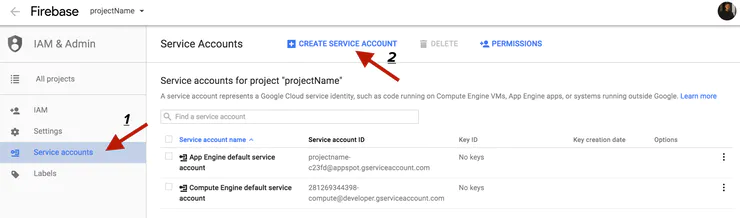
Step 2: Create a new project.



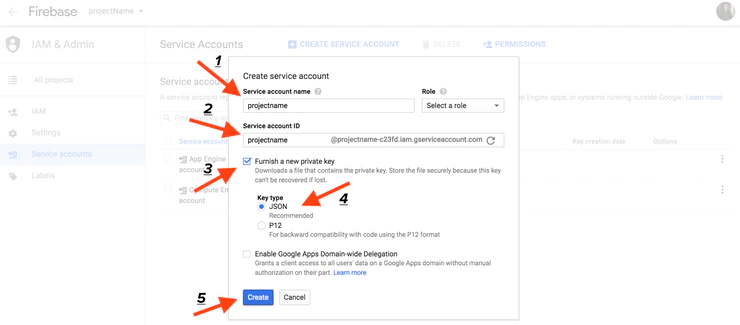
Step 3: Create a firebase security profile.



After accessing permissions, click on 'Service Accounts' (1) then click 'Create Service Account' (2).

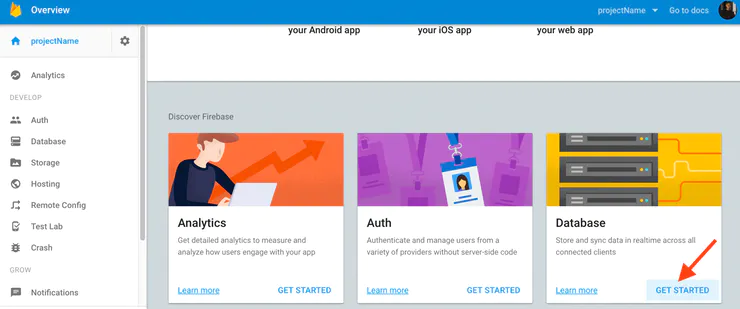


A window should show up giving you some options to create your security profile. Start by first setting your service account name (1). Next name your Service account ID (2). Then check 'Furnish a new private key' (3) and choose JSON format (4). Click create when finished (5).



Once created, the security profile will download. Place this file in the GitHub repository you have downloaded and re-name it something easy to remember.

Step 4: Configure your Real-Time Database. Start by clicking on 'getting started' in your firebase console.



Next, we have to setup the Real-Time DB. This can easily be done by importing this JSON (can be found in GitHub repository under db.json)

{

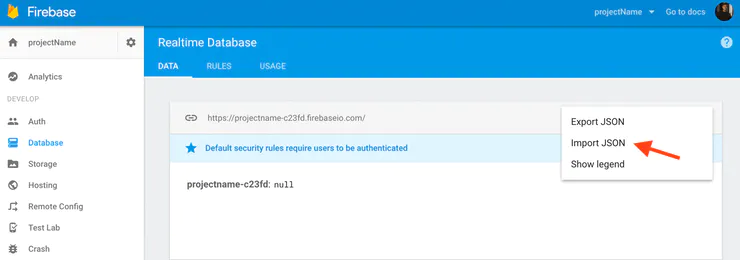
"Alexa" : {

"Read" : "",

"Write" : ""

}

}



**Setting up Repository Code:**

In index.js, change the APP\_ID variable to your Amazon App ID found on the Alexa section of the developer console,

var APP\_ID = '<Amazon APP ID HERE>';

In firebase.js and database.js, change the firebase.initializeApp function parameters databaseURL to your firebase Real-Time database URL (found on the firebase console) and serviceAccount parameter to the saved security profile you created above (.json format).

firebase.initializeApp({

databaseURL: '<firebaseURLhere>',

serviceAccount: './<firebaseSecurityProfile>.json'

});

Create a developer account at Kairos and replace the app\_id and app\_key parameters in firebase.js to the credentials on your Kairos developer account.

headers:{

'Content-Type': 'application/json',

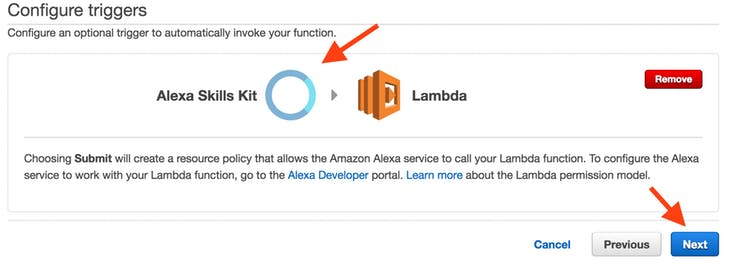
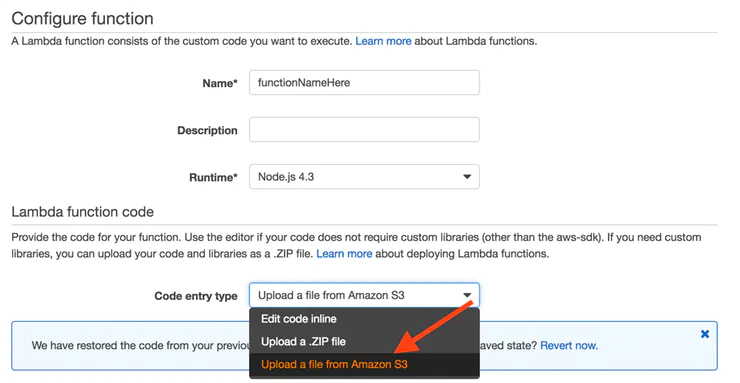
'app\_id': '<KairosAppID>',

'app\_key': '<KairosAppKey>'

}

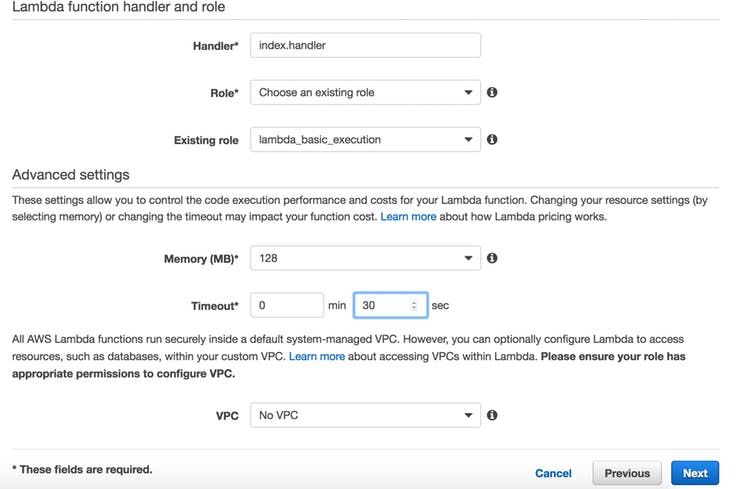
Now, zip up the files to get ready to push it to lambda. Zip index.js, database.js, responses.js, <SecurityProfile>.json, node\_modules, AlexaSkill.js, and package.json. Upload zipped file to your Amazon S3 bucket. (to upload to lambda, the zip file is large).

**Setting up Alexa and Lambda:**

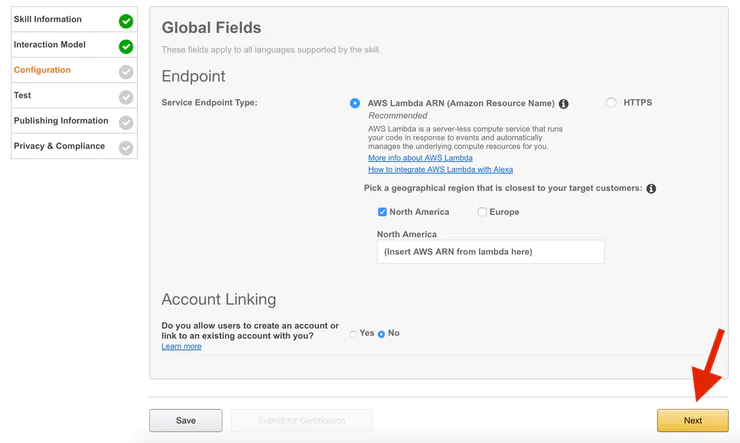
* Create your lambda function 
* Setup Alexa Skills Kit trigger and click next. 
* Next, name your function and choose to upload from Amazon S3 (large ZIP file). 

From the uploaded zip from before, find its S3 link and copy it into the text input field shown.

Fill in the rest of the fields as follows:



**Setting up the skill to test on the development console:**

* Go to <https://developer.amazon.com/home.html> and sign on with your AWS account. Once logged in, Click on the Alexa tab.
* Next, click on the Add a New Skill Button.
* Then, fill out the input fields as follows below and click Next.
* Add the contents from “IntentSchema.txt” to the Intent Schema input text.
* Add the contents from “SampleUtterances.txt” to the Sample Utterances input text and press the next button.
* Fill in the Global Fields as shown below. 

**Testing and enjoying skill:**

On the Raspberry Pi terminal, type:

node firebase.js

Make sure the camera is setup on your door (or in-front of your face to test it). On the Amazon Developer Console, where your Alexa application exists. Copy your Lambda ARN into the created Alexa skill and begin testing. Follow the flow shown in the VUI to test the skill to its extent.