1).<https://firebase.google.com/docs/database/>

* Firebase is a NoSQL cloud Realtime database where data can be saved.
* The data in Firebase is stored in JSON type and can be sync in realtime to the connected clients.
* Firebase provides platform for building cross-platform apps by using iOS, Javascript and Android SDKs.Along with this it provides Realtime Database instance for read/write of data to the connected clients and can also receive updates of latest data through data synchronisation.

2). <https://sites.ndtv.com/roadsafety/important-feature-to-you-in-your-car-5/>

* In 2013, around 134K people died in India due to road accidents which is comparatively higher than the deaths of soldiers.
* Everyday 1214 road accidents takes place out of which everyday 377 people die.
* Every 4 min. 1 person killed out of serious road accidents.
* Delhi city is at the peak in terms of road accidents.

3). <https://www.makeuseof.com/tag/how-self-driving-cars-work-the-Cloudant>

* Google announced in 2008 of its new Autonomous cars to avoid accidents of around 30,000 people every year in United States.
* The Autonomous Car uses following components:
* LIDAR (Laser Illuminating Detection and Ranging):-It is placed on top of the car to generate 3D map and get 360 degree view by using Velodyne-64 beam laser to see hazards by calculating the distance and profile of that object.
* Radar:-LIDAR can’t measure the speed of neighbouring cars so Radar was used for it. 2 radar sensors are placed in front bumper and 2 in the rear it to send signals to the processor for applying brakes or get into clear way, so as to avoid potential accidents.
* Sonar- It allows the car to effectively cross-reference data from other systems in real time to apply the brakes, pre-tension seat belts for impact, or swerve to avoid obstacles.
* Mapping:- Google uses its own [map system](https://www.makeuseof.com/tag/technology-explained-google-maps-work/), as well as GPS satellites, inertial measurement units, and a wheel encoder to determine actual speed. The system works alongside the on-board cameras to process real-world information as well as GPS data

4). [https://becominghuman.ai/building-an-image- classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8](https://becominghuman.ai/building-an-image-%20classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8)

* Building a Convolutional Neural Network to train out of the 10K images of dogs and cat and will able to predict a particular given image if it is a dog or cat.
* Keras-adeep learning library in python is used to build CNN(Convolutional Neural Network).
* They involves 4 major steps: Convolution, Pooling, Flattening ,Full connection.

**Python code for Image capture by raspberry pi , Image Processing and Image Upload on Firebase Cloud by put() method and get data from Firebase Cloud by using get() method**

def Capture():

import picamera

with picamera.PiCamera() as camera:

camera.resolution = (320, 240)

camera.capture('img.jpg')

def imgPro():

import cv2

img = cv2.cvtColor(cv2.imread('img.jpg'), cv2.COLOR\_BGR2GRAY)

cv2.imwrite('img1.jpg', img)

def Encode():

import base64

with open('img1.jpg', "rb") as imageFile:

str = base64.b64encode(imageFile.read())

return str

def upload(stack, imgCode):

from firebase import firebase

firebase = firebase.FirebaseApplication('https://cars-7b9d5.firebaseio.com/', None)

path = '/'+stack+'/Last'

last = firebase.get(path, None)

result = firebase.put(stack, last, {'imgCode':imgCode})

last = str(int(last) + 1)

last = firebase.put(stack, 'Last', last)

Capture()

print('=======================================================')

imgPro()

upload('Left', Encode())

print(Encode())

**Python code to control the Car by Raspberry pi**

import time

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

# 17, 18 are RIGHT motor

# 22, 23 are LEFT motor

GPIO.setup(17, GPIO.OUT)

GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.OUT)

GPIO.setmode(GPIO.BCM)

GPIO.setup(22, GPIO.OUT)

GPIO.setmode(GPIO.BCM)

GPIO.setup(23, GPIO.OUT)

def forward():

print('forward')

GPIO.output(18,GPIO.LOW)

GPIO.output(22,GPIO.LOW)

GPIO.output(17,GPIO.HIGH) #forward

GPIO.output(23,GPIO.HIGH) #forward

time.sleep(1)

GPIO.output(17, GPIO.LOW) #stop

GPIO.output(23, GPIO.LOW) #stop

time.sleep(0.5)

def backward():

print('back')

GPIO.output(17, GPIO.LOW)

GPIO.output(23, GPIO.LOW)

GPIO.output(18, GPIO.HIGH)

GPIO.output(22, GPIO.HIGH)

time.sleep(1)

GPIO.output(18, GPIO.LOW)

GPIO.output(22, GPIO.LOW)

time.sleep(0.5)

def tleft():

print('left')

GPIO.output(18,GPIO.LOW)

GPIO.output(22,GPIO.LOW)

GPIO.output(23,GPIO.LOW)

GPIO.output(17,GPIO.HIGH) #LEFT

time.sleep(2)

GPIO.output(17, GPIO.LOW) #stop

time.sleep(0.5)

def tright():

print('right')

GPIO.output(17,GPIO.LOW)

GPIO.output(18,GPIO.LOW)

GPIO.output(22,GPIO.LOW)

GPIO.output(23,GPIO.HIGH) #RIGHT

time.sleep(3)

GPIO.output(23,GPIO.LOW) #stop

time.sleep(0.5)

def main():

GPIO.output(17, GPIO.LOW)

GPIO.output(18, GPIO.LOW)

GPIO.output(22, GPIO.LOW)

GPIO.output(23, GPIO.LOW)

forward()

time.sleep(0.25)

backward()

time.sleep(0.25)

tleft()

time.sleep(0.25)

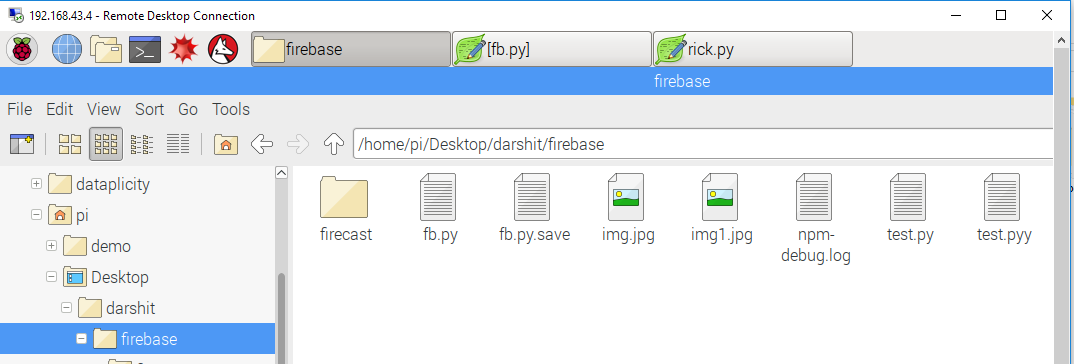
tright()

time.sleep(0.25)

GPIO.cleanup()

main()

GPIO.cleanup()



**Socket Programming code (Earlier implementation)**

from socket import \*

serverName = 'Dexter1902'

serverPort = 12000

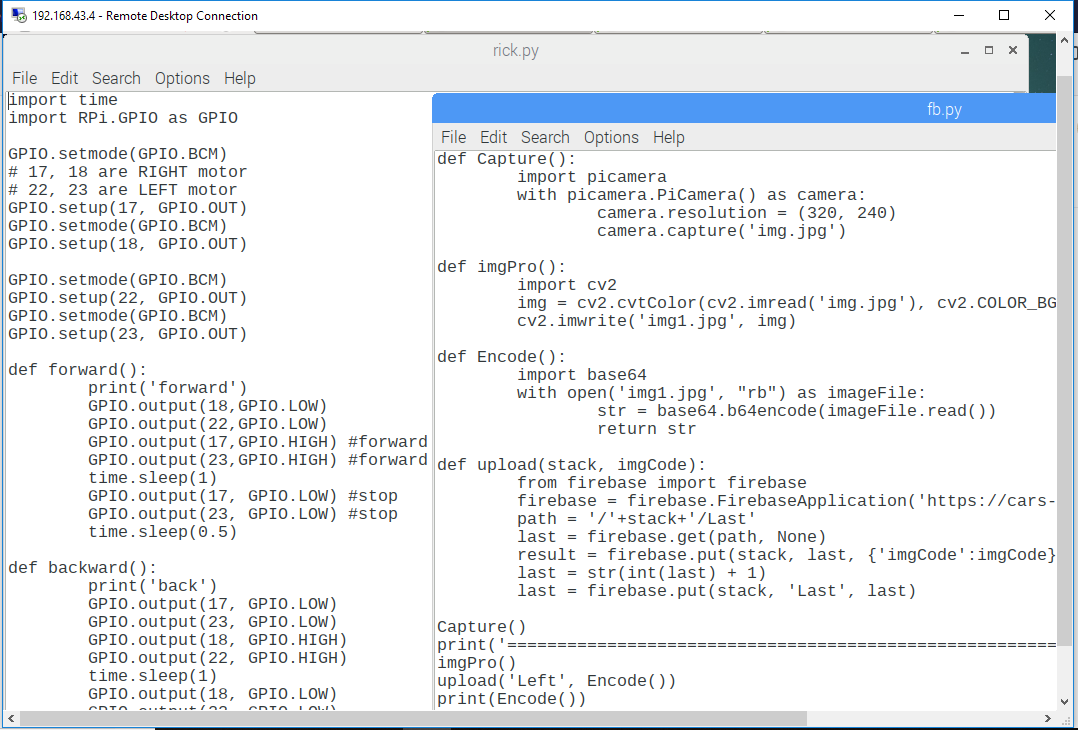
clientSocket = socket(AF\_INET,SOCK\_STREAM)

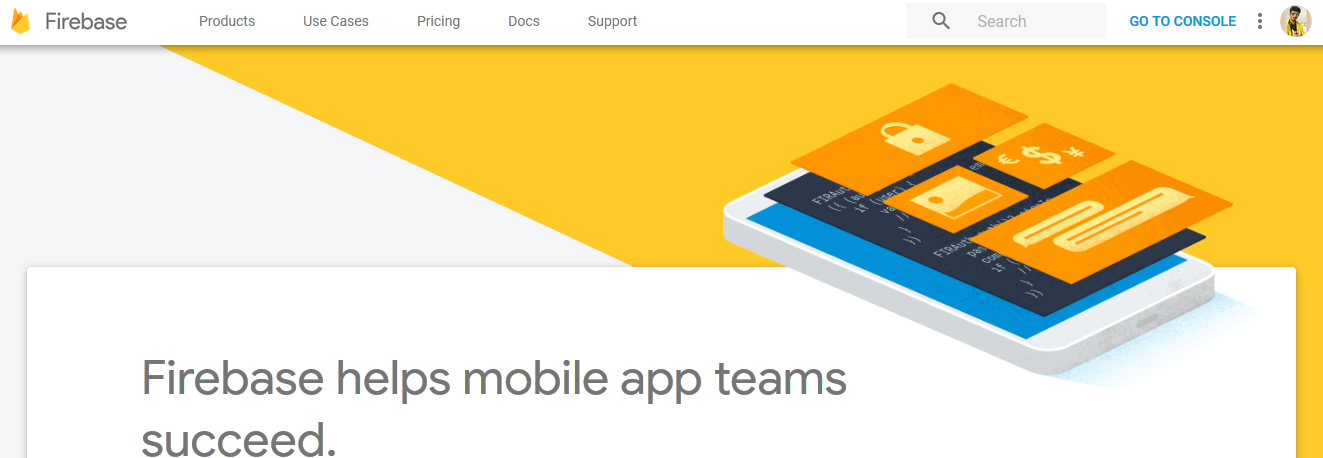
clientSocket.connect((serverName,serverPort))

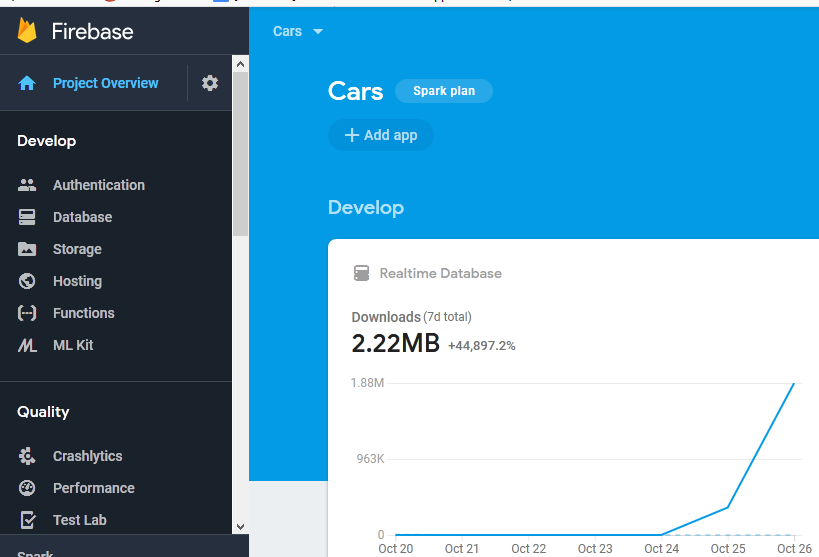
sentence = input('Enter :').encode()

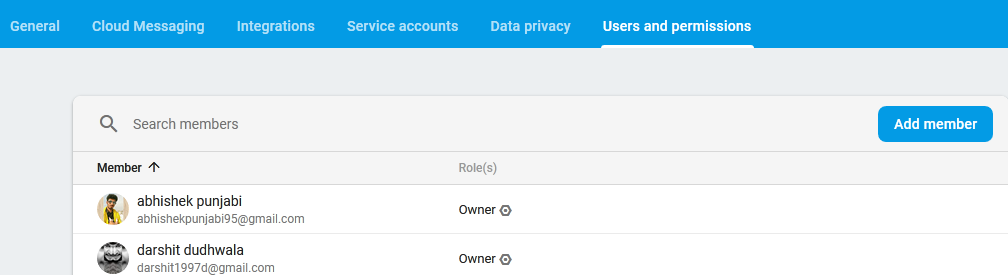
clientSocket.send(sentence)

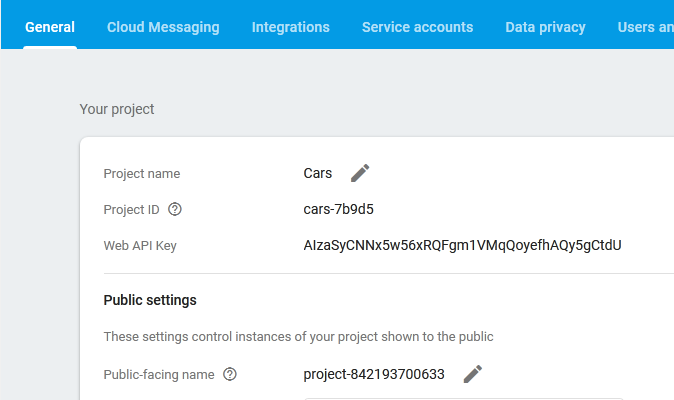
clientSocket.close()







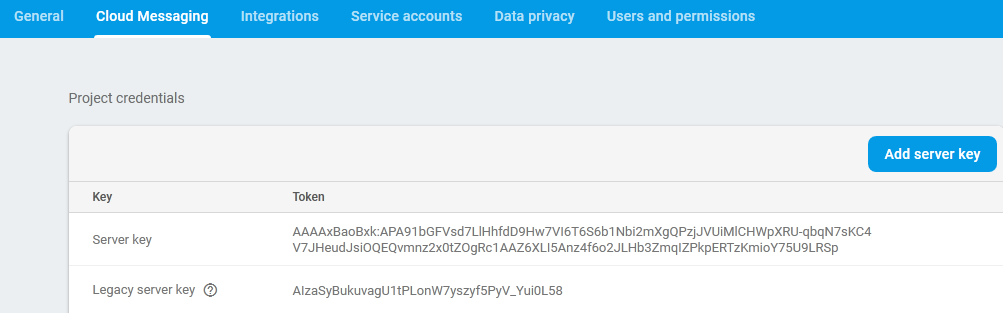




Project name Cars

Project ID cars-7b9d5

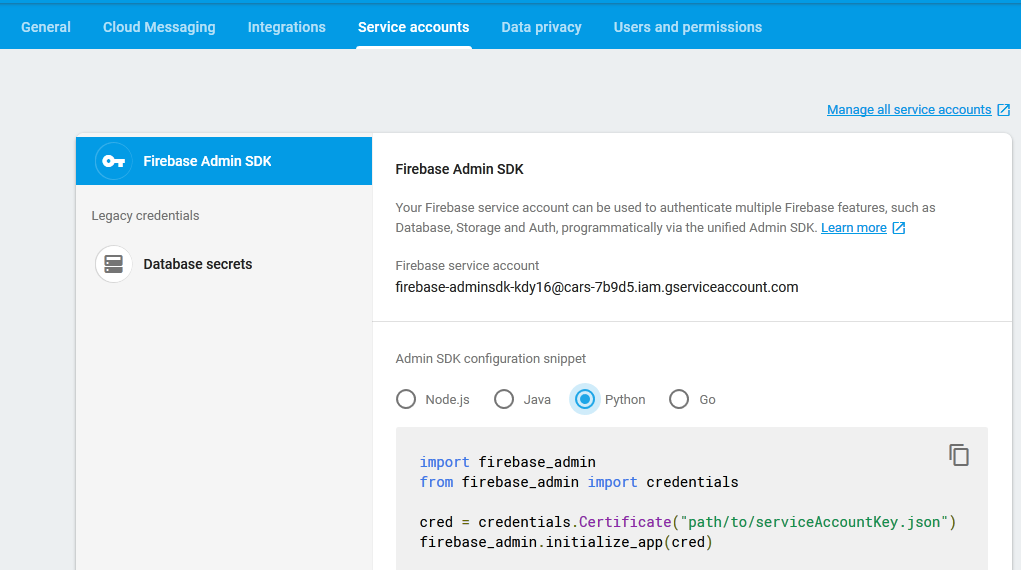
Web API Key AIzaSyCNNx5w56xRQFgm1VMqQoyefhAQy5gCtdU



Server Key AAAAxBaoBxk:APA91bGFVsd7LlHhfdD9Hw7VI6T6S6b1Nbi2mXgQPzjJVUiMlCHWpXRU-qbqN7sKC4V7JHeudJsiOQEQvmnz2x0tZOgRc1AAZ6XLI5Anz4f6o2JLHb3ZmqIZPkpERTzKmioY75U9LRSp

Legacy Server Key AIzaSyBukuvagU1tPLonW7yszyf5PyV\_Yui0L58

Sender ID 842193700633



Firebase service account

firebase-adminsdk-kdy16@cars-7b9d5.iam.gserviceaccount.com

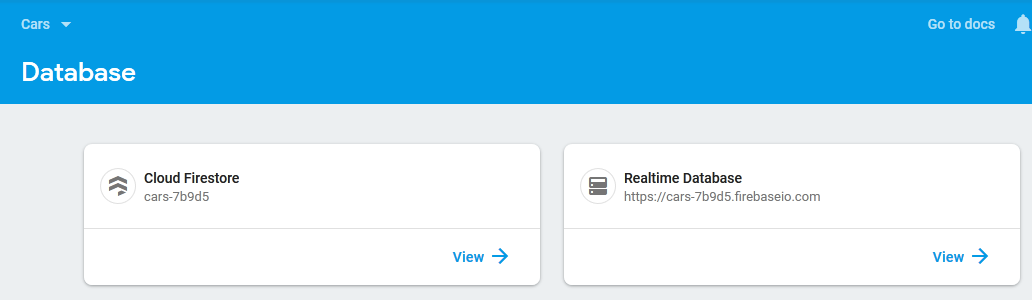
**Admin SDK configuration Python**

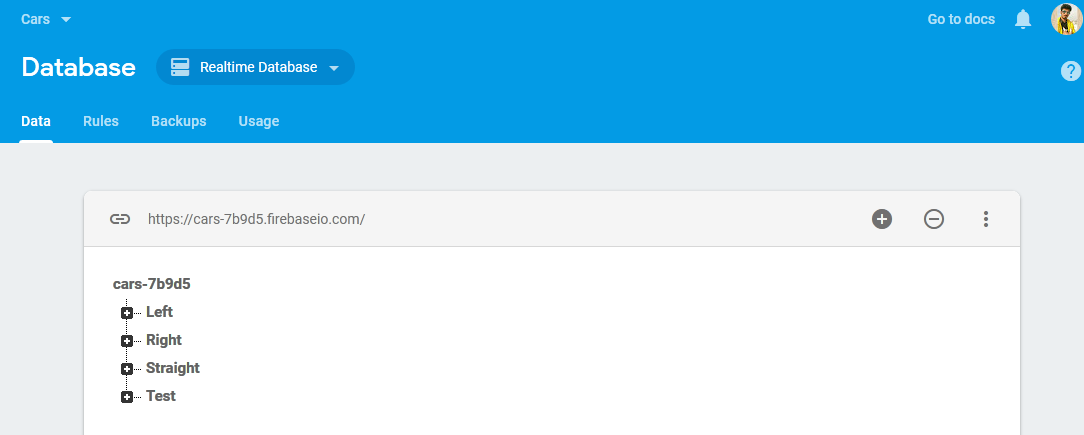
import firebase\_admin

from firebase\_admin import credentials

cred = credentials.Certificate("path/to/serviceAccountKey.json")

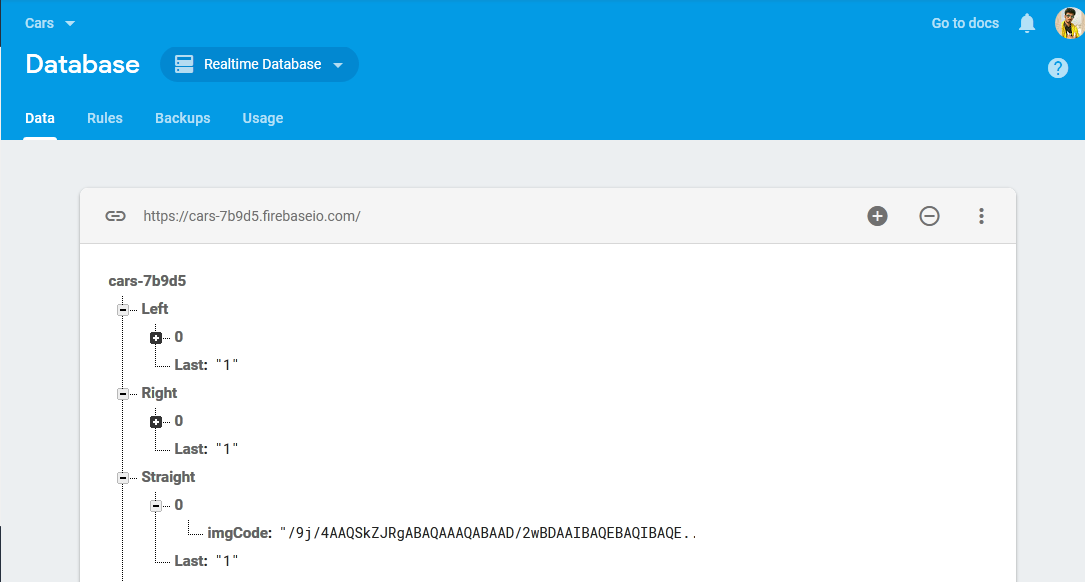
firebase\_admin.initialize\_app(cred)



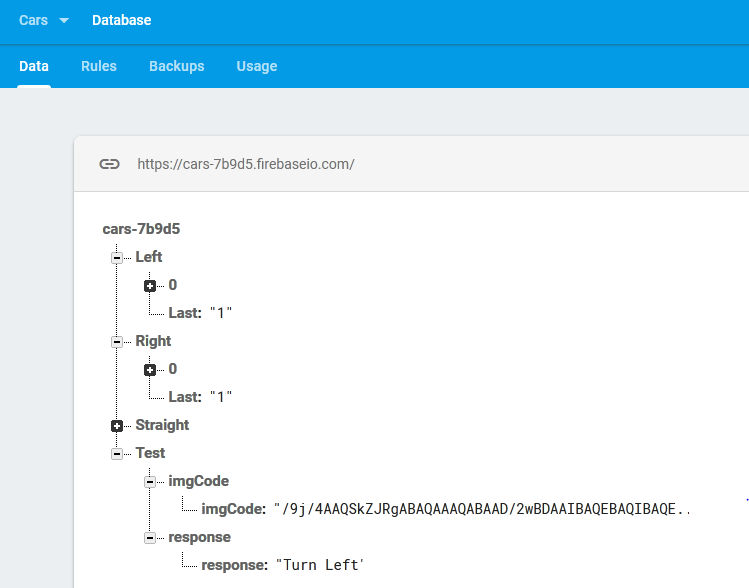


<https://cars-7b9d5.firebaseio.com/>

**Training Data set have 3 stacks Left Right and Straight**



**Test image uploaded on Firebase**

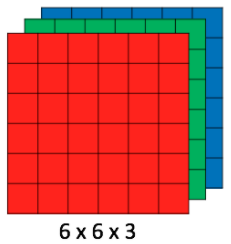


# CNN Convolutional Neural Network

<https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d( h = Height, w = Width, d = Dimension ). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.



**Array of RGB Matrix**

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.



**Neural network with many convolutional layers**

**Convolution Layer**

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

**Pooling Layer**

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

* Max Pooling
* Average Pooling
* Sum Pooling

Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

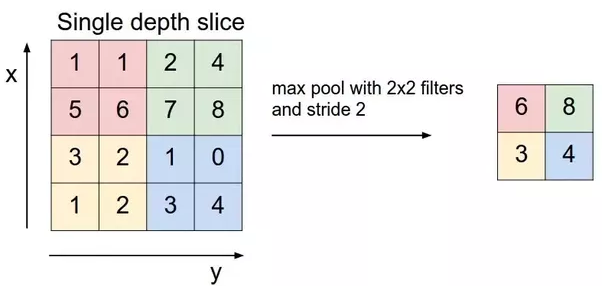
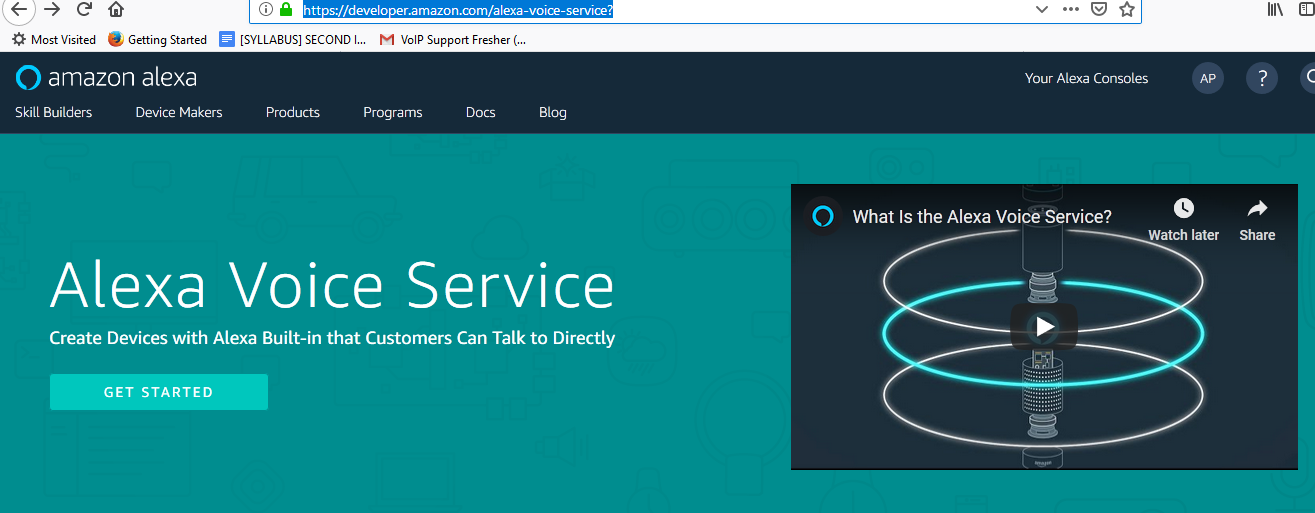


Figure 8 : Max Pooling

<https://developer.amazon.com/alexa-voice-service>?



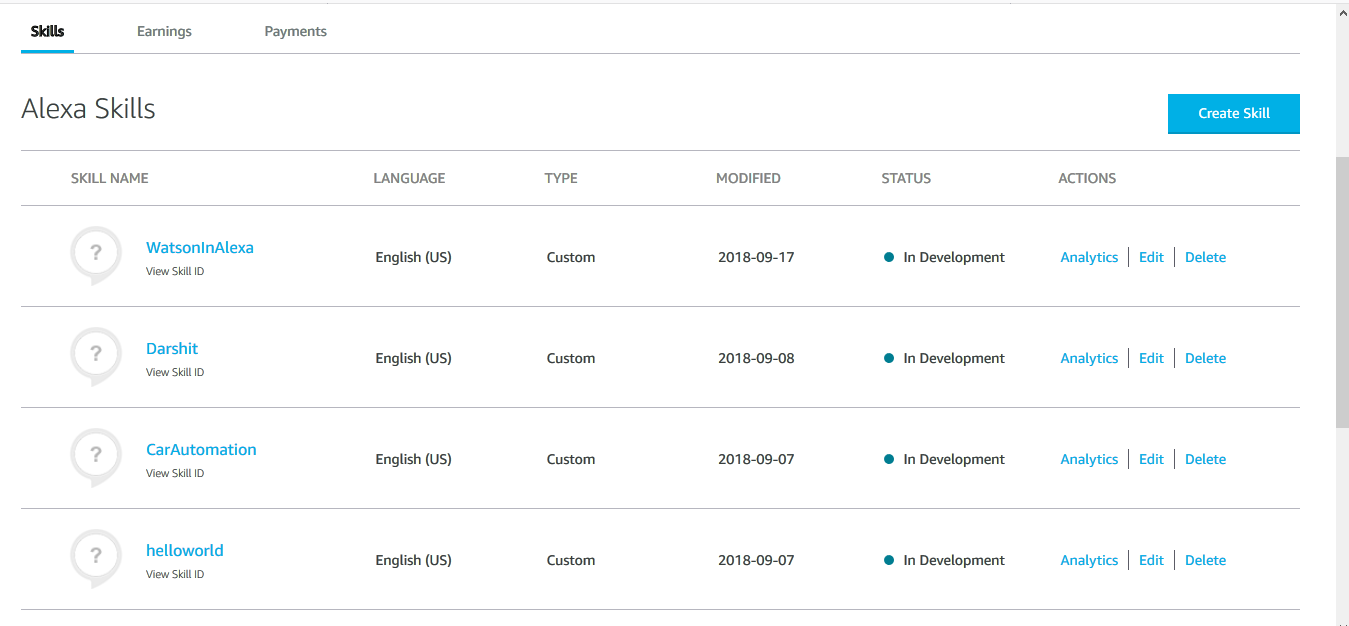
## What Is an Alexa Built-in Product?

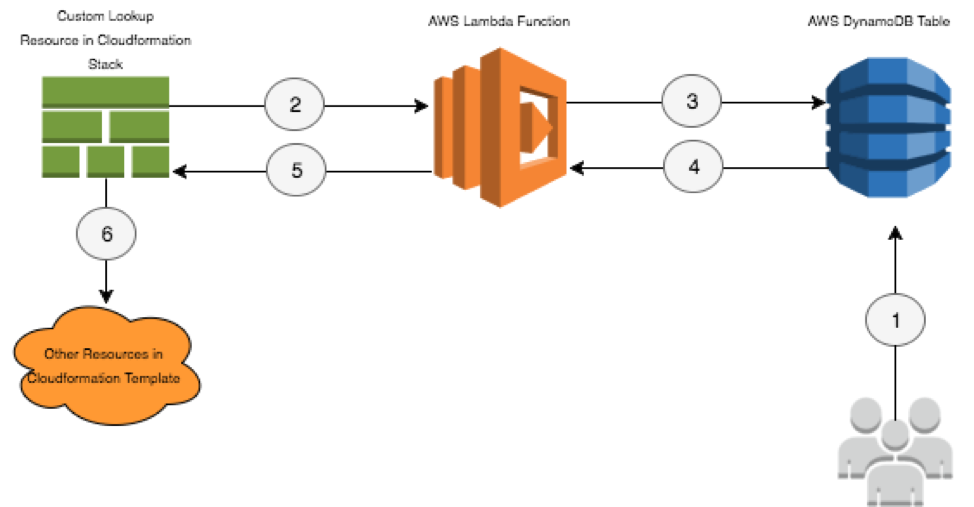
Alexa built-in is a category of devices created with the Alexa Voice Service that have a microphone and speaker. You can talk to these products directly with the wake word “Alexa,” and receive voice responses and content instantly. Alexa built-in products work with Alexa skills and Alexa-compatible smart home devices, bringing familiar capabilities from the Amazon Echo family of devices to a range of new form factors and use cases developed by leading brands.

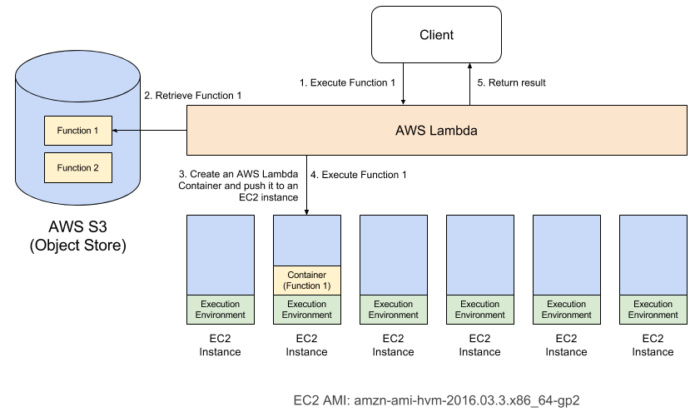
## Why Alexa Voice Service?

The Alexa Voice Service (AVS) enables you to access cloud-based Alexa capabilities with the support of AVS APIs, hardware kits, software tools, and documentation. We simplify building voice-forward devices with Alexa built-in by handling complex speech recognition and natural language understanding in the cloud, reducing your development costs and accelerating your time to market. Best of all, regular Alexa updates bring new features to your device and add support for a growing assortment of compatible smart home devices

**Alexa Skills set at developer account**







AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running. With AWS Lambda, you can run code for virtually any type of application or backend service - all with zero administration. AWS Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging.

* Amazon Elastic Compute Cloud (Amazon EC2) service offers flexibility and a wide range of EC2 instance types to choose from. It gives you the option to customize operating systems, network and security settings, and the entire software stack, but you are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.
* Elastic Beanstalk offers an easy-to-use service for deploying and scaling applications onto Amazon EC2 in which you retain ownership and full control over the underlying EC2 instances.

<https://in.mathworks.com/help/deeplearning/index.html>

Deep Learning Toolbox™ provides a framework for designing and implementing deep neural networks with algorithms, pretrained models, and apps. You can use convolutional neural networks (ConvNets, CNNs) and long short-term memory (LSTM) networks to perform classification and regression on image, time-series, and text data. Apps and plots help you visualize activations, edit network architectures, and monitor training progress.

<https://in.mathworks.com/help/fuzzy/index.html>

<https://searchenterpriseai.techtarget.com/definition/fuzzy-logic>

**Fuzzy logic** is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean **logic** on which the modern computer is based.

Natural language (like most other activities in life and indeed the universe) is not easily translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in [binary](https://whatis.techtarget.com/definition/binary) terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing.) It may help to see fuzzy logic as the way reasoning really works and binary or Boolean logic is simply a special case of it.

Fuzzy logic includes 0 and 1 as extreme cases of truth (or "the state of matters" or "fact") but also includes the various states of truth in between so that, for example, the result of a comparison between two things could be not "tall" or "short" but ".38 of tallness."

Fuzzy logic seems closer to the way our brains work. We aggregate data and form a number of partial truths which we aggregate further into higher truths which in turn, when certain thresholds are exceeded, cause certain further results such as motor reaction. A similar kind of process is used in [neural networks](https://searchenterpriseai.techtarget.com/definition/neural-network), [expert systems](https://searchenterpriseai.techtarget.com/definition/expert-system) and other [artificial intelligence](https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence) applications.