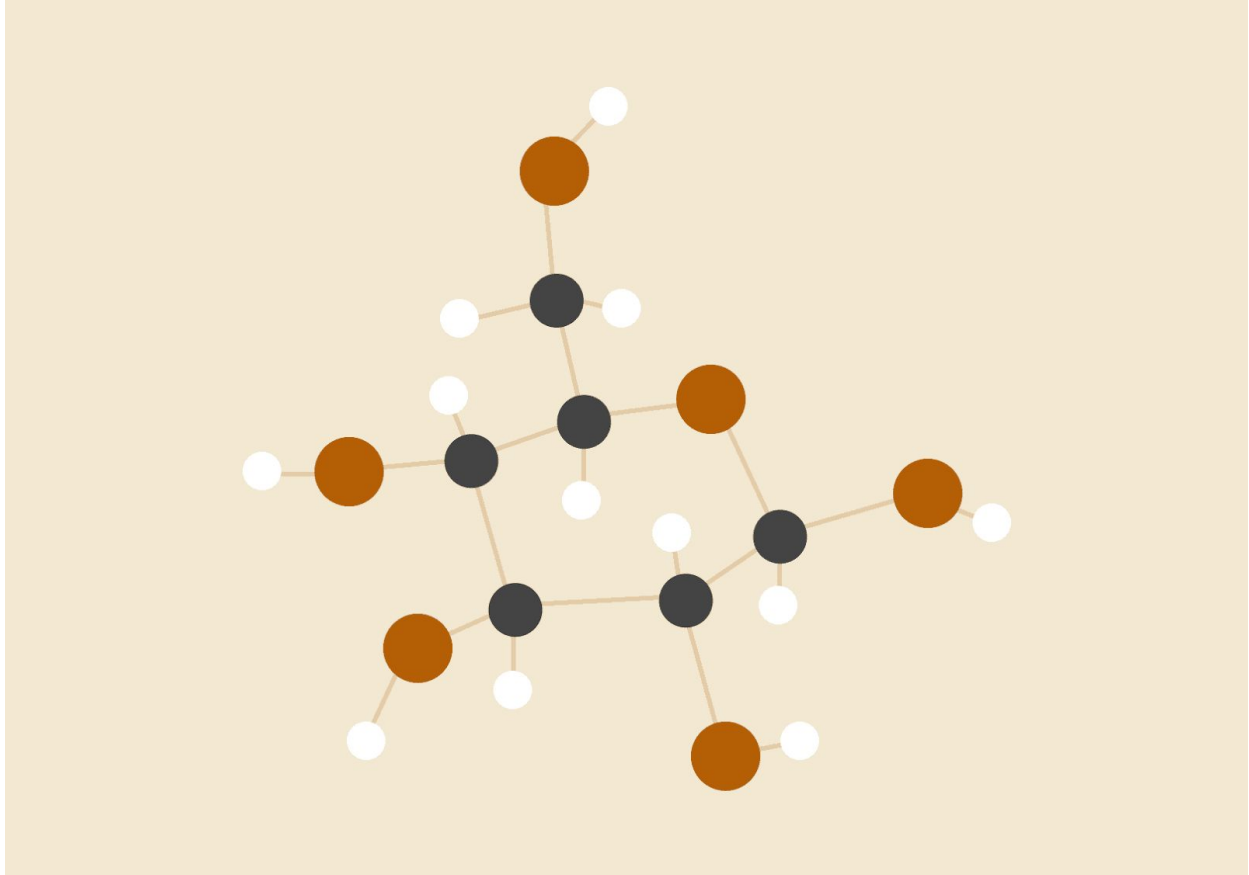# Caption AI

Project report

Vrajesh Mehta (VAM345)

Avi Patel (APP440)

Agam Parekh (ADP511)

Karan Katliwala (KRK386)

INTRODUCTION

We are creating a web service in which you upload an image and you will get the relevant and witty captions for that picture. Everything is done by machine intelligence so that is why we call it 'CaptionAI'. Every minute millions of images are uploaded on social platforms like Instagram, Twitter, Facebook, etc. All images are uploaded with hashtags and caption with it. Captions have become the new way to show your wittiness on social media and to impress others. Finding a perfect caption is a very tedious task. Here we come with best captions scrapped from all over the internet.

SOLUTION

To give the most possible relevant caption suggestions we do the following.
For that we do the following :

- Decide the most popular 50 tags on Instagram
- Machine Learning Model
  - Get 1000 Images for each of that tag using web-scraping from the internet (specifically from https://www.pexels.com/ in our case)
  - Get labels for all of those thousands of images from Google's Cloud Vision API
  - Map them and with image's link and send to the AWS
  - Train preprocessed data in Sagemaker on AWS
- Provide Captions
  - Give caption suggestion scrapped from www.brainyquote.com

ARCHITECTURE

We are going to use two cloud services:-

- Google Cloud Platform
  - Cloud Vision API
  - IoT hub
  - GCP Pub/Sub
  - Google Cloud Function

- Amazon web services
  - API Gateway
  - Lambda
  - Sagemaker
  - DynamoDB
  - Data Pipeline

Data collection:

This part is done in **GCP**.
We first created one repository in IoT core of GCP, then we added 10 virtual IoT devices in that registry.

We created the Pub-Sub topic "image-links" to get data from IoT devices and to drive it for further process.

Each IoT devices web-scrapes image links from the prexel.com and send to the Pub-Sub topic "image-links".

We created Google Cloud Function with the trigger of Pub-Sub, which will get executed as soon as any data comes in "image-links". This Cloud function basically does two many things.

- Get labels for an image from Cloud Vision API
- Store it into the AWS's DynamoDB
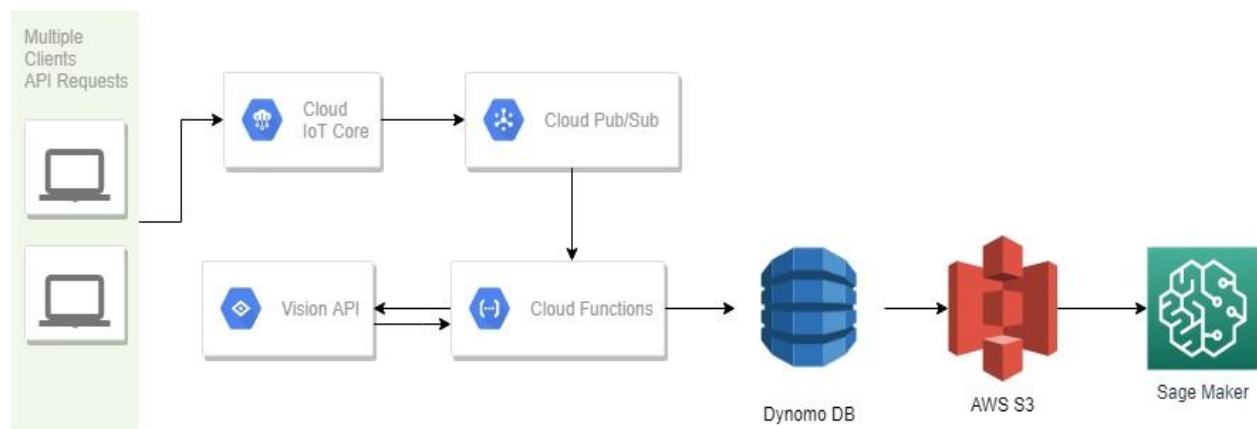
Data Pipeline:

This is where **AWS** comes in the picture:

With the help of Amazon EMR Cluster, we have created a data pipeline.

Every 7 days it gets all the data from dynamoDB table and stores it into the s3 bucket in JSON format. Once we have the data we retrain redeploy our neural network. (we want to automate this part in the future)

Exporting Data from DynamoDB to Amazon S3
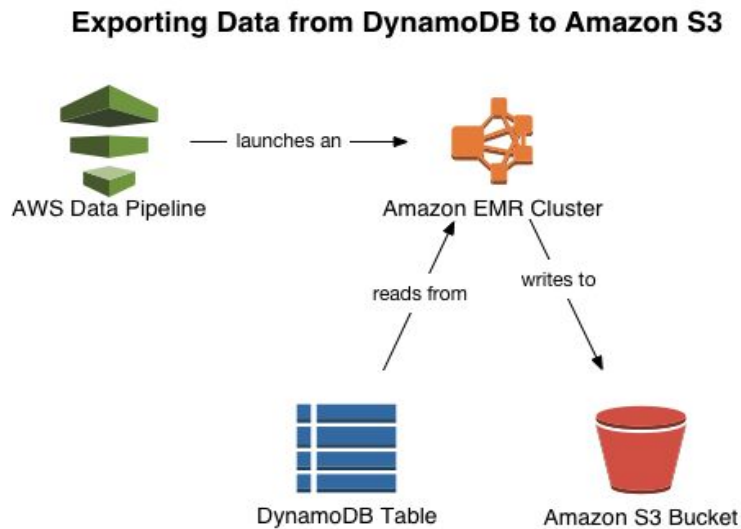
User Interaction:
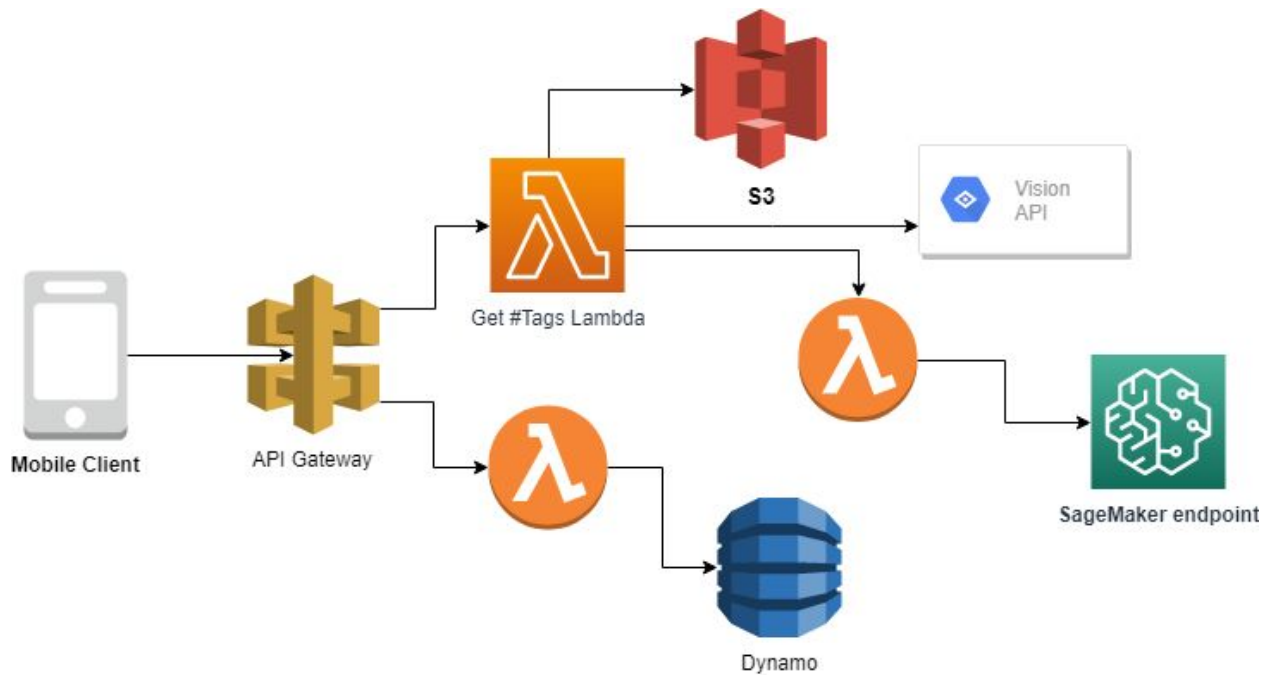
The frontend application interacts with our system through an API gateway. The user uploads an image for which they want to get the hashtags and captions. For every image, we store it in S3 bucket and fetch the labels along with the confidence from GCP Vision API. Once we have the labels and their confidence we get the relevant hashtags by running our pre-trained model in Sagemaker. The user can then select any hashtag and we present them with relevant captions for that hashtag.

ScreenShot

LINKS

Demo: https://s3.amazonaws.com/proj-frontend/index.html

Github: https://github.com/vrajesh978/cloud-final-project

Video: https://www.youtube.com/watch?v=2fiQou7y07Y&feature=youtu.be