**Question – 2**

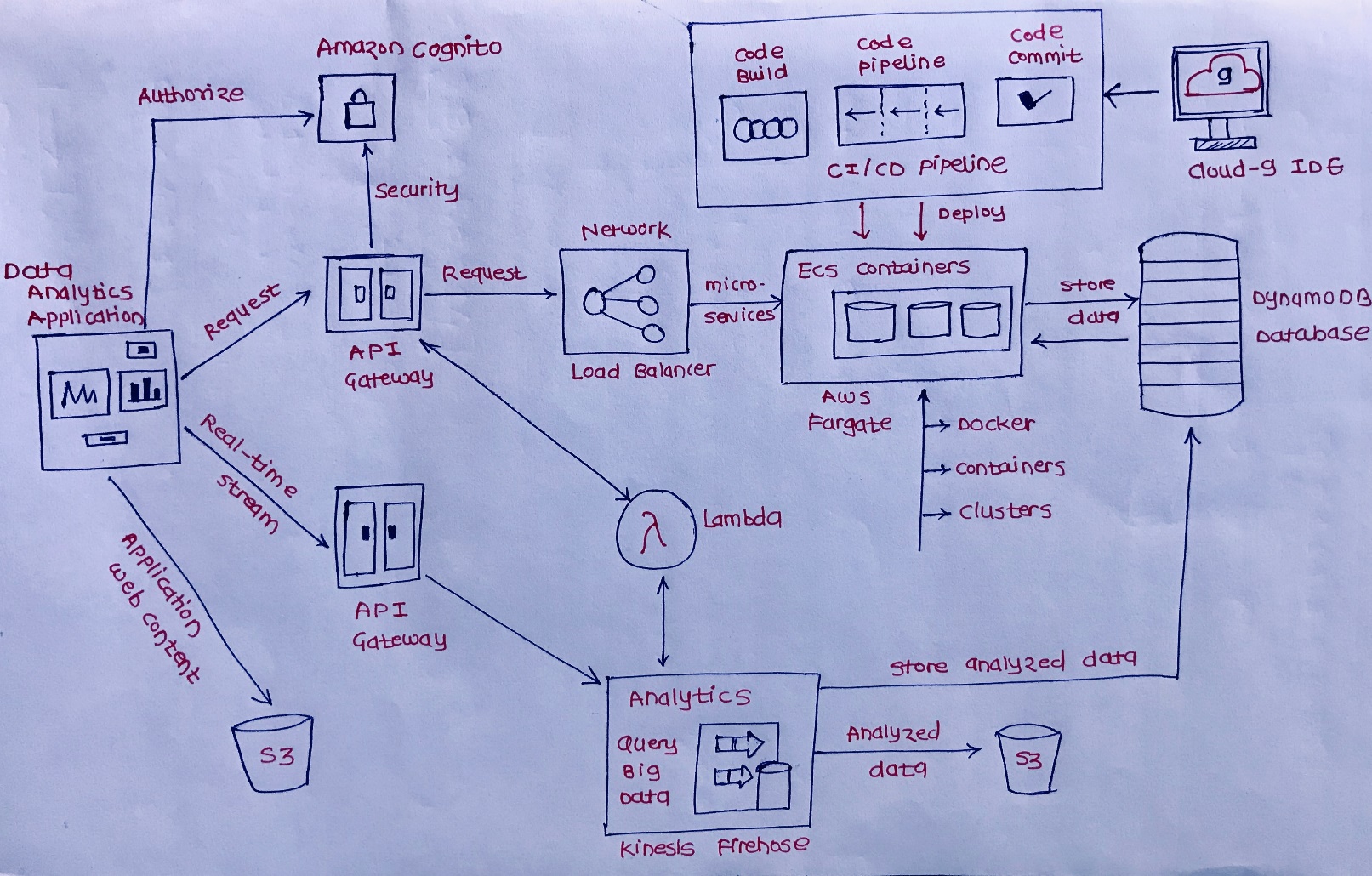
**What are some of the interesting, exciting, or challenging technical problems you are working on?**

As part of my Master Study, I was working with one of my Cloud Computing professors on an interesting project – **Multi-Tenancy Bigdata Analytics on Modern Serverless Cloud Platform (AWS)**

[**https://github.com/vinaykhedekar21/AWS-Modern-Web-Application**](https://github.com/vinaykhedekar21/AWS-Modern-Web-Application)

**Project Goals/challenges:**

* Building architecture of serverless modern web application on the AWS platform which mimics the way industry has serverless modern web application architecture running on a cloud platform.
* Developing multi-tenant (two) applications on the same platform and analyzing two different bigdata sets for different matrix generated by multi-tenant web applications.
* The academic goal was to learn the AWS services and get hands-on experience in implementing maximum service to build the designed architecture.
* Being honest, my personal goal was to learn and get hands-on experience of AWS cloud platform as implementing industry level architecture was a bit challenging task for us.



**Technical Challenges-**

* Mimic an infrastructure same as what industry has for serverless web applications (shown in the architecture diagram)
* Creating infrastructure components using the command-line tool leads to preparing complex YAML scripts.
* Synchronizing all individual infrastructure components as a whole architecture

**Infrastructure setup on AWS Cloud Platform**

1. Setup Core Infrastructure

a) Setup Amazon Virtual Private Cloud (VPC)

b) Setup NAT API Gateways with 4 subnets

c) Write CloudFormation scripts (YAML/JSON)

d) Setup AWS Cloud watch log group

e) Setup security group

f) Setup user roles and policies

2. Setup Clusters and Containers

a) Setup an Elastic Container Service (ECS)

b) Deploy a service with AWS Fargate

c) Create AWS Fargate cluster

d) Create Network Load balancers

e) Create ECS and Docker service

f) Setup Elastic Container Registry (ECR)

g) Write a script to deploy Docker containers

h) Configure services to communicate with the load balancer

3. Setup a database system

a) Create NoSQL DynamoDB Database

b) Create a Database schema and Indexes

c) Create user roles and policies

**Application Programming Interfaces (API)**

1. Develop Micro-services

a) Develop a Flask framework based micro-services and API’s using python, AWS Lambda functions, and DynamoDB

b) Create Docker Images and deploy to the Docker container.

2. Automate CI/CD Processes

a) Write CloudFormation script to setup CI/CD services.

b) Setup AWS CodeCommit service

c) Setup AWS CodeBuild Service

d) Setup AWS CodePipeline Service

e) Deploy application code Docker image to clusters on the Fargate container.

**Results:**

The project is still in the progress as it requires many efforts as a student to learn and then try implementing it. Personal results are I have gained hands-on experience and knowledge of AWS services.