The Student Plaza

By : Rinkle Rani (W1590085), Himani Vaidya(W1629165), Abhilash H. Srivathsa (W1606870) & Aanchal Rai (W1606869)

Problem Statement

- **The Student Plaza** is a platform for students in a university (SCU) to discover, buy or sell items to each other.
- This would create a marketplace for only students/faculty in a university so they can safely interact with each other and work towards a more sustainable environment through reusing items such as books, chairs, tables, etc.

Motivation For This Project

- To **learn** as many cloud computing concepts that were taught in class.
- To implement technologies that are **currently relevant in the industry.**
- To build a cloud-based application with seamless integration and **auto-scaling feature.**

- A lot of marketplace apps are **publicly available**. Travelling to the buyers house is **not a safe option**.
- An average college student ends up **discarding 640 pounds** of their belongings annually when they move out. Many of these items can be reused by the incoming students.

Background Information

EKS: Service designed to run Kubernetes on AWS for creating, managing, scaling containerized applications.

On demand cloud computing platform provided by Amazon that includes AWS S3, EKS, Lambda and many more services.

ElasticSearch: Managed Service providing full text search, log analytics, real-time application monitoring which works on high volumes of **unstructured** data

Rds: Distributed relational database provided by AWS designed to simplify setup, scaling of database for use of applications

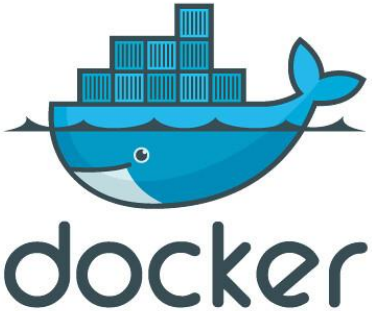
DynamoDB: Nosql database supporting document & key-value formats designed for fast scaling

Related Work

Facebook has a Market Place feature where users buy / sell items to the public whether its old or new. Method of product delivery is upto the buyer and seller.

Another example where clients can buy & sell on a platform.

Docker

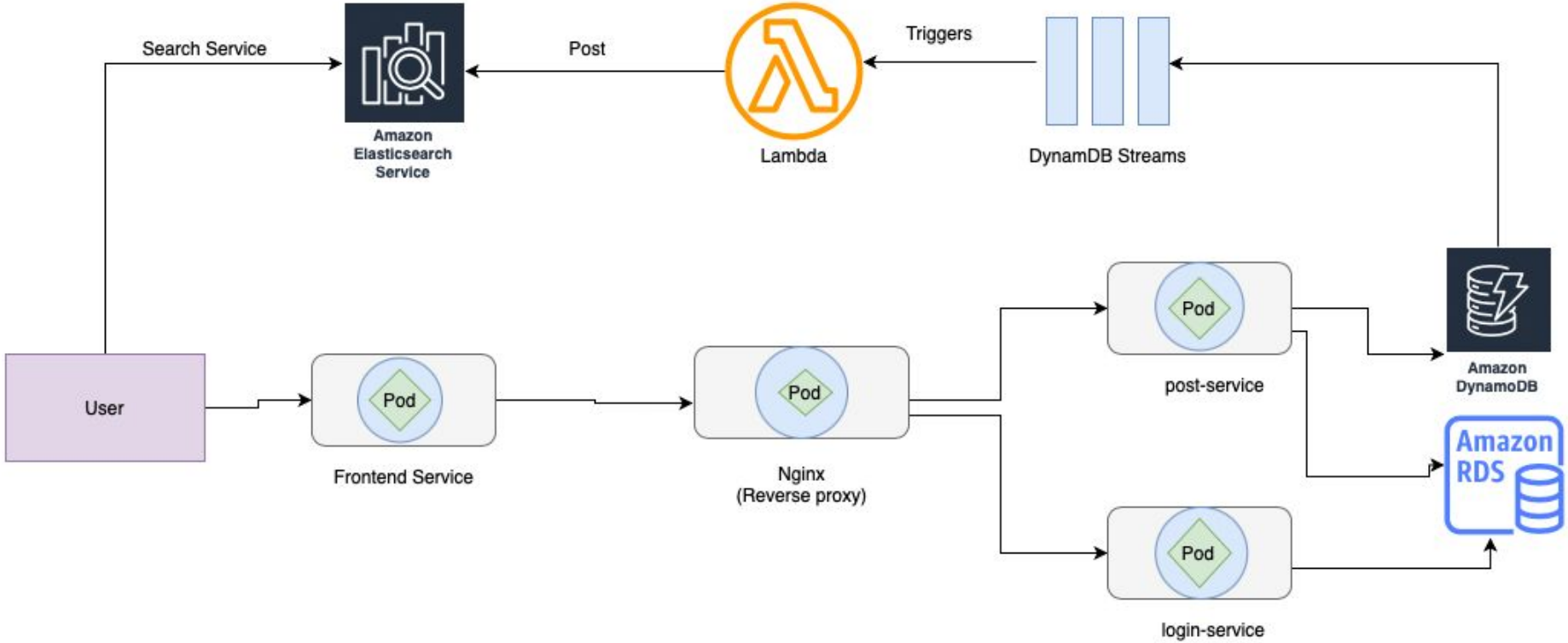


- Used Docker to make our microservices containerized.
- Docker is open source platform to develop, ship and run applications.
- EKS fetches the docker images from docker hub and deploy automatically

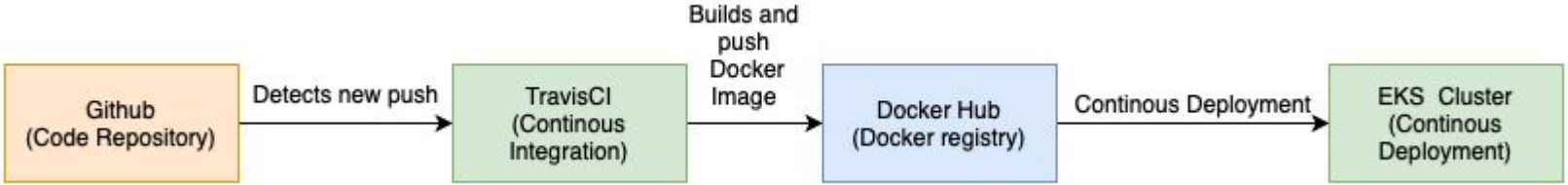
CloudFormation

- AWS service that uses template files to automate the setup of AWS resources
- CloudFormation stack: EKS Cluster, Worker nodes, Control planes, Security groups, DynamoDB, RDS, Elastic Search.
- It is easy to deploy/redeploy the stack with just one command.

System Architecture

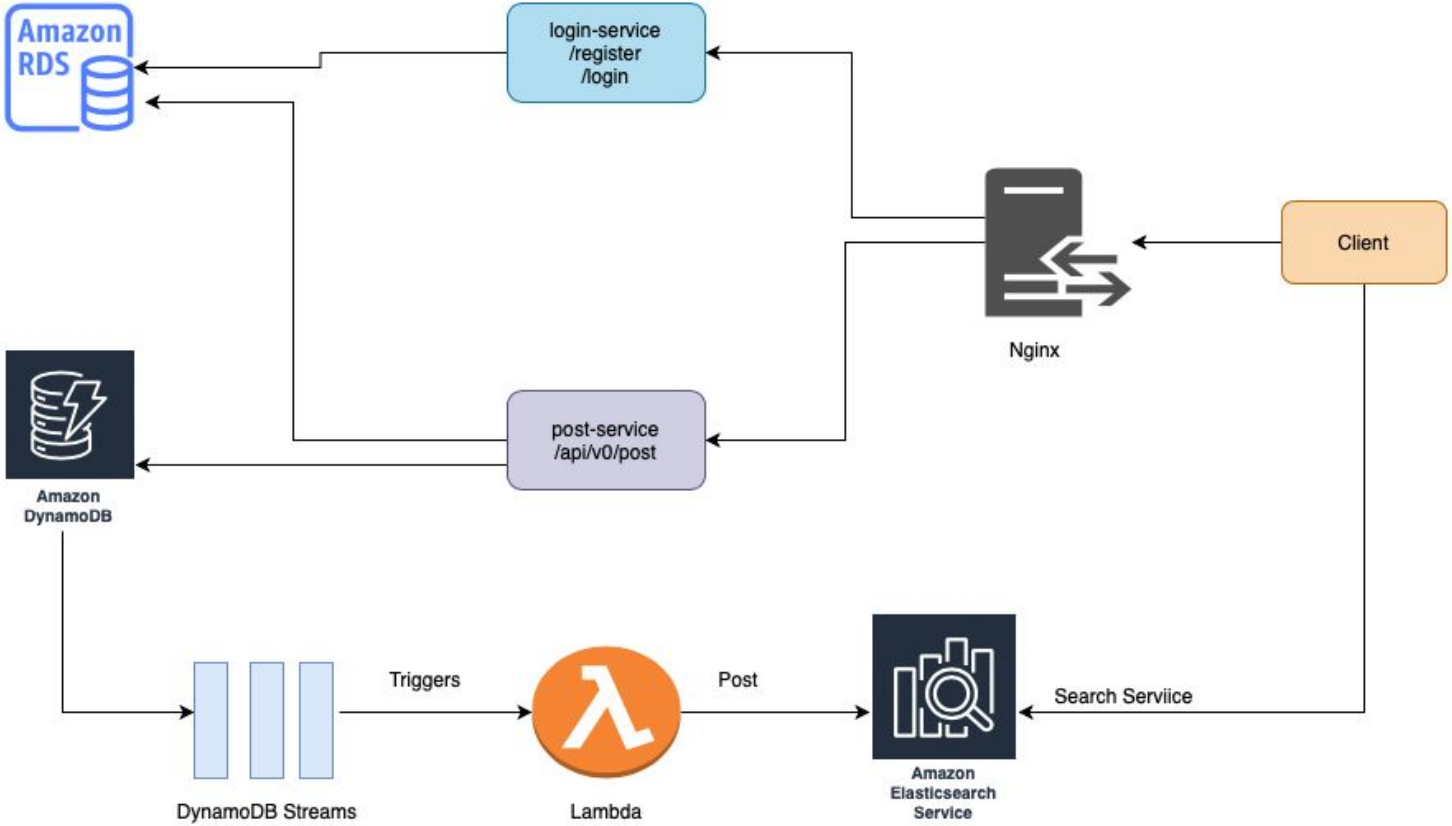


CI/CD Pipeline

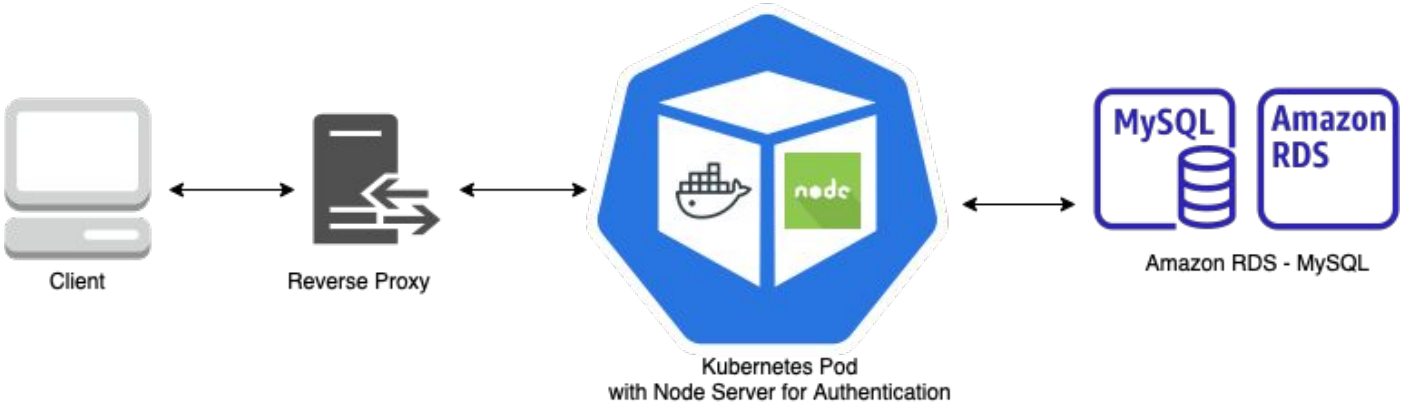


- Implemented Continuous Integration using Travis CI
- Implemented Continuous Deployment using Elastic Kubernetes Services
- Whenever new push made on Github, TravisCI will automatically build new Docker Images from the new Code and push the images to Docker hub.

Software Architecture Overview



Microservice - Login & Signup



- Client will send a request to either signup or login
- Reverse Proxy will direct it to the Authentication Service
- The Auth microservice is implemented using Node and Express
- It is running in a Kubernetes Pod as a Docker image
- The user details are stored in a MySQL database instance on AWS RDS

Microservice - Post-Service

- A containerized microservice which supports all the CRUD operations for a Post.
- Uses DynamoDB to store and fetch all the posts.
- Interacts with RDS MySql for authentication purposes.
- Supports Api versioning
- TechStack: Typescript, Node.js, Express.js, Docker, CloudFormation

Microservice - Frontend

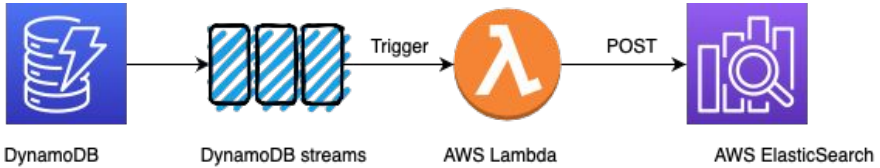
- A containerized microservice frontend developed for student plaza application, supporting features like create post, search specific post, view all posts, create user account.

- Integrated with login, signup API endpoints for authentication purpose, with elasticsearch API for search of specific post and with post API endpoints for add post and get all posts operations

- TechStack: ReactJS

Search Service

- DynamoDB Streams captures item-level modifications in any DynamoDB table and stores this information in a log for up to 24 hours.
- Whenever there is something new in DynamoDB Streams, it will trigger our lambda function with those records.
- Lambda then POSTs the data to ElasticSearch
- Search queries are sent to ElasticSearch DB endpoint via POST requests.



Outcome

- Implemented a marketplace which is exclusively for Students of Santa Clara University
- The entire application is on the AWS cloud and completely scalable
  - We were able to implement many of the concepts taught in this course!
- Helps to reduce waste and reuse items
- Beneficial for the students
- As the application is exclusive, it improves on other available apps as the buyer and seller are both trustworthy

Possible Future Work

- Post Api supports all the CRUD operations, we can add these functionalities to the frontend to support post deletion and updation.
- Image support in Post Api.
- Security Features like enhanced Authentication and Authorization
- Search recommendations
- Filters for search

DEMO


Thank you!