PARSHVANATH CHARITABLE TRUST'S

A.P. Shah Institute of Technology

Thane, 400615

Academic Year: 2022-23

Department of Computer Engineering

CSL605 SKILL BASED LAB COURSE: CLOUD COMPUTING

Mini Project Report

- **Title of Project** : AWS-Themepark
- **Year and Semester** : T.E. (Sem VI)
- **Group Members Name and Roll No.** : Janhavi Silaskar(46)
  Atharva shinde (37)
  Anuj Shinde (36)
  Vikram Shetty (34)

Table of Contents

Problem Definition

The purpose of our project is to develop a simple web application on AWS which shows a website of a theme-park having rides. The webpage displays the information about the rides, has a feature of changing the display rides. The webpage can also display the live ride wait timings, the data used in simulation can be used in Amazon Quick sight, allowing data analysis.

This project can be further sub-divided into different modules like hosting the web page, creating a database, and connecting the database with the front-end, adding the ride-wait timing feature, translation feature and data stats . This web application provides a convenient and user-friendly way for users to view am easy-to-use webpage, giving information about the theme par. Various AWS services like Amplify, Lambda, Dynamo DB, Kinesis Firehose, Quicksights will be used to develop and deploy this application.

Introduction

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon. It offers a wide range of cloud-based services that can be used to build, deploy, and manage web applications and services. AWS provides a scalable and flexible infrastructure that enables businesses and organizations to provision resources on-demand and pay only for what they use quickly and easily. AWS has a pay-as-you-go pricing model, which means that users only pay for the resources they use and can easily scale up or down as required. AWS also provides a range of tools and services to help users manage and monitor their applications, as well as support for various programming languages and frameworks. In this project, we will be using various AWS services such as Amplify, DynamoDB, and API Gateways to develop and deploy a simple web application .These services help develop secure, scalable, and reliable applications easily. We deployed the web page using Amplify which provides a public DNS to access our website from anywhere. For each of the modules in the project, different AWS services were used. The cloud9 IDE provided by AWS was used to add any changes in the code. DynamoDb has been used as a Database for storing information. The Lambda function is also used in the feature where the real time waiting for the ride is calculated. Amazon Kinesis Data Firehose is a fully managed service that reliably loads streaming data into data lakes, data stores and analytics tools. It can capture, transform, and load streaming data into Amazon S3, which is one of the important services used in this project. The data collected into the S3 bucket in then used by the quicksight service, allowing user to have an analysis of the visitors of the park. HTML, CSS and JavaScript have been used to build the web page and make it responsive.

Description

AWS services used in the project-

1] **Amplify:**

AWS Amplify is a development platform and set of tools provided by Amazon Web Services (AWS) that allows developers to build scalable and secure web and mobile applications quickly and easily. We can also use Amplify to integrate APIs with other AWS services, such as Lambda and DynamoDB. Amplify provides a range of storage options, including Amazon S3, Amazon DynamoDB, and Amazon Aurora, which make it easy to store and retrieve data from your application. Amplify provides built-in authentication and authorization capabilities that make it easy to secure your application and protect user data. AWS Amplify also supports a range of programming languages and frameworks, including React, Angular, Vue, iOS, and Android.

2] **Lambda:**

AWS Lambda is a serverless compute service provided by Amazon Web Services (AWS) that allows us to run code in response to events or triggers, without the need to provision or manage servers. Lambda allows us to run our code in the cloud, without the need for a physical server, which can help you reduce costs and increase scalability. AWS Lambda can be integrated with a range of other AWS services, including Amazon S3, Amazon DynamoDB. AWS Lambda provides built-in security features, including encryption of data in transit and at rest, and can be configured to integrate with AWS Identity and Access Management (IAM) to manage access to resources.

3] DynamoDB:

Amazon DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS). DynamoDB is a fully managed service, which means AWS takes care of managing the infrastructure, backups, and software updates, so you don't have to worry about managing your own database servers. DynamoDB provides several security features, including encryption at rest and in transit, fine-grained access control using AWS Identity and Access Management (IAM), and integration with Amazon VPC.

4] API Gateway:

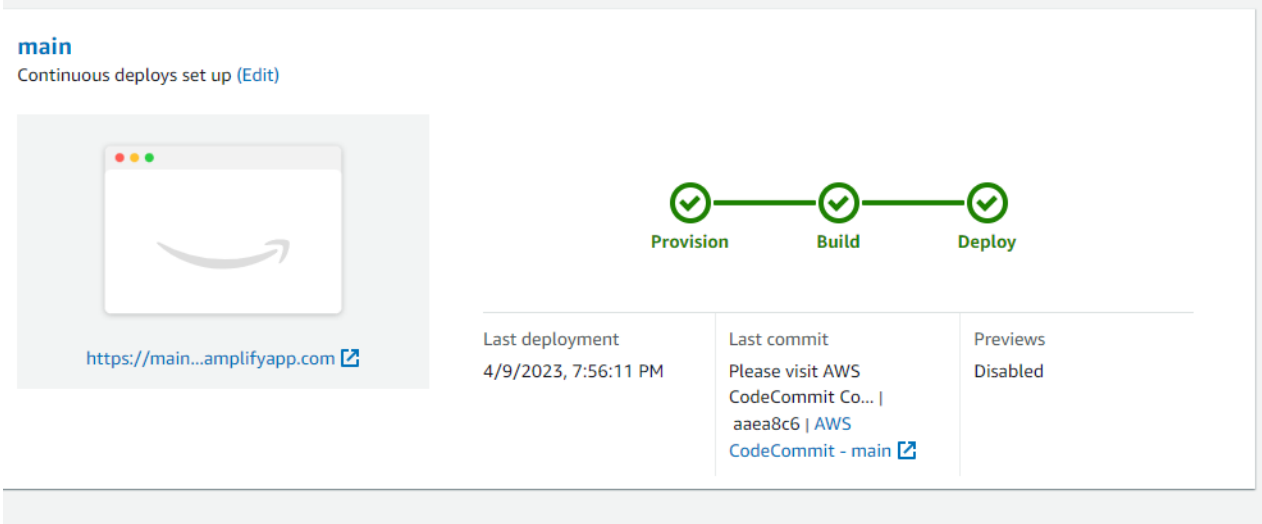
Amazon API Gateway is a fully managed service provided by Amazon Web Services (AWS) that makes it easy for developers to create, publish, and manage APIs at any scale. It allows you to create RESTful APIs, WebSocket APIs, and HTTP APIs that enable backend services to communicate with clients in a secure and scalable manner. API Gateway provides several security features, including authentication and authorization using AWS Identity and Access Management (IAM). API Gateway integrates with a wide range of AWS services, including AWS Lambda, Amazon S3, and Amazon DynamoDB. This allows you to create APIs that access data and services from other AWS services.

5] Kinesis Firehose:

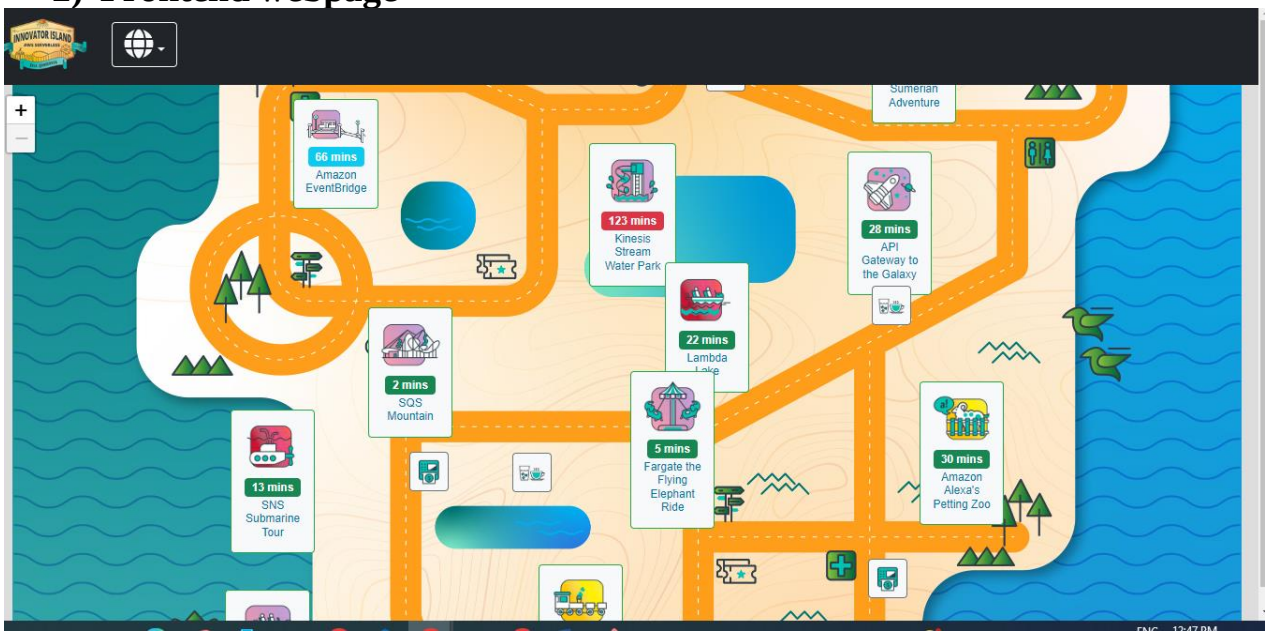
Amazon Kinesis Data Firehose is an extract, transform, and load (ETL) service that reliably captures, transforms, and delivers streaming data to data lakes, data stores, and analytics services. It automatically scales to match the throughput of your data and requires no ongoing administration. You can configure a delivery stream and start sending data from hundreds of thousands of data sources to be loaded continuously to AWS – all in just a few minutes.

Implementation details with screen-shots

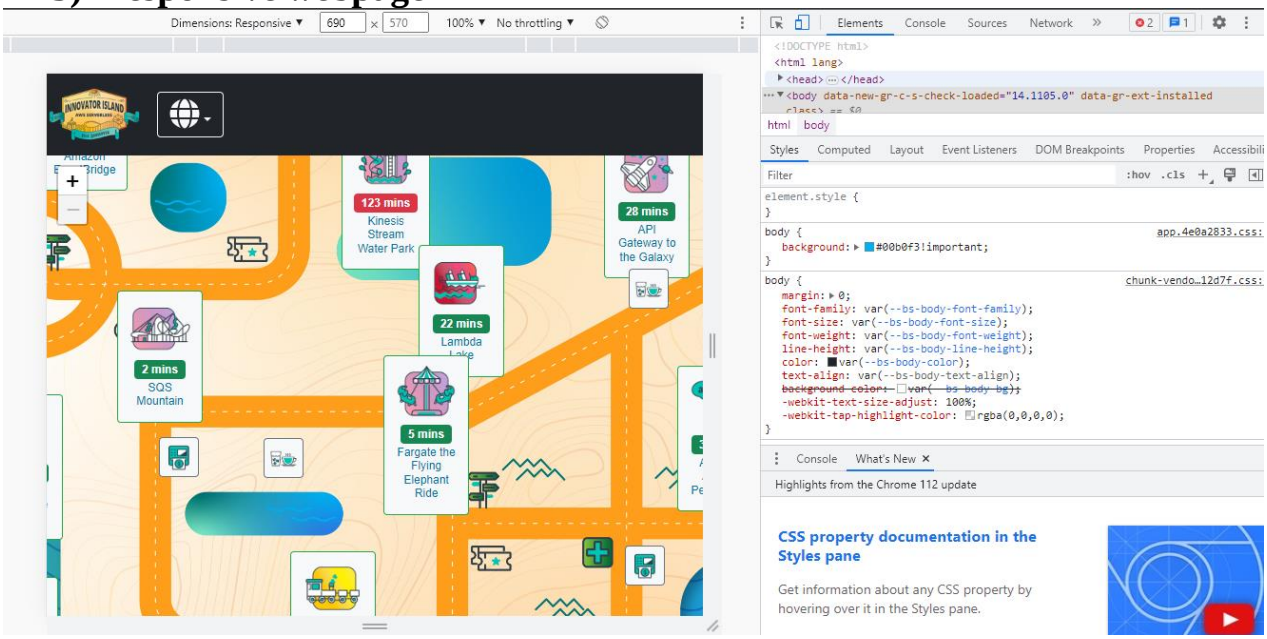
1) Hosting with Amplify



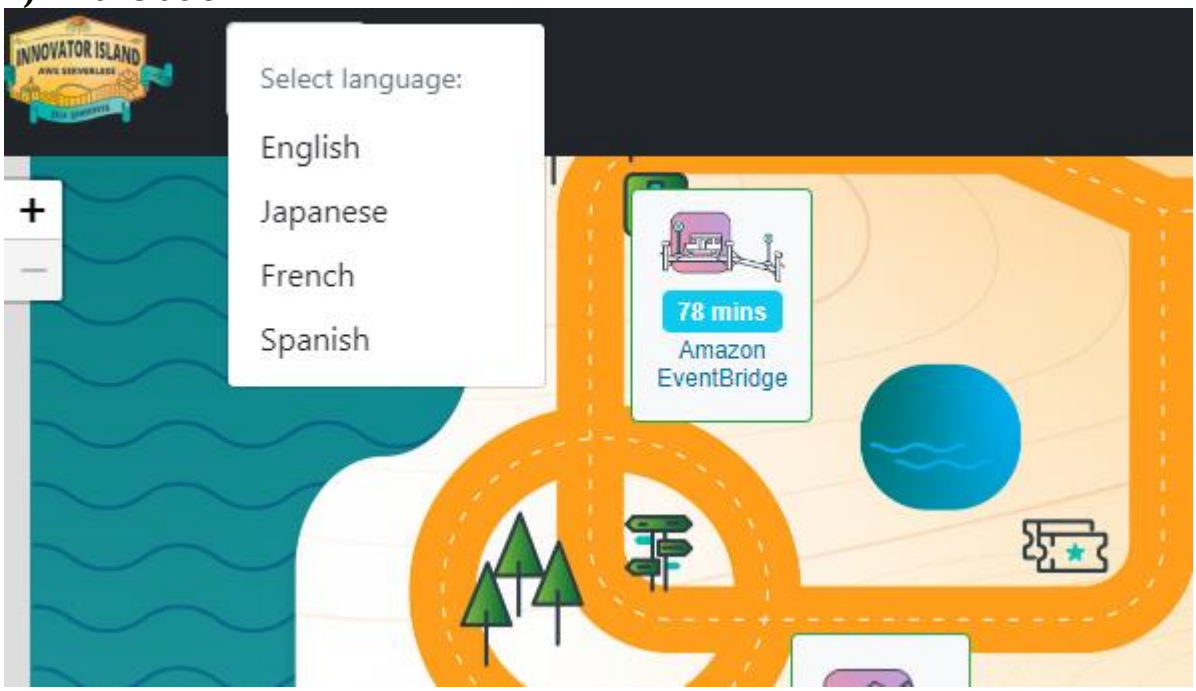
2) Frontend webpage



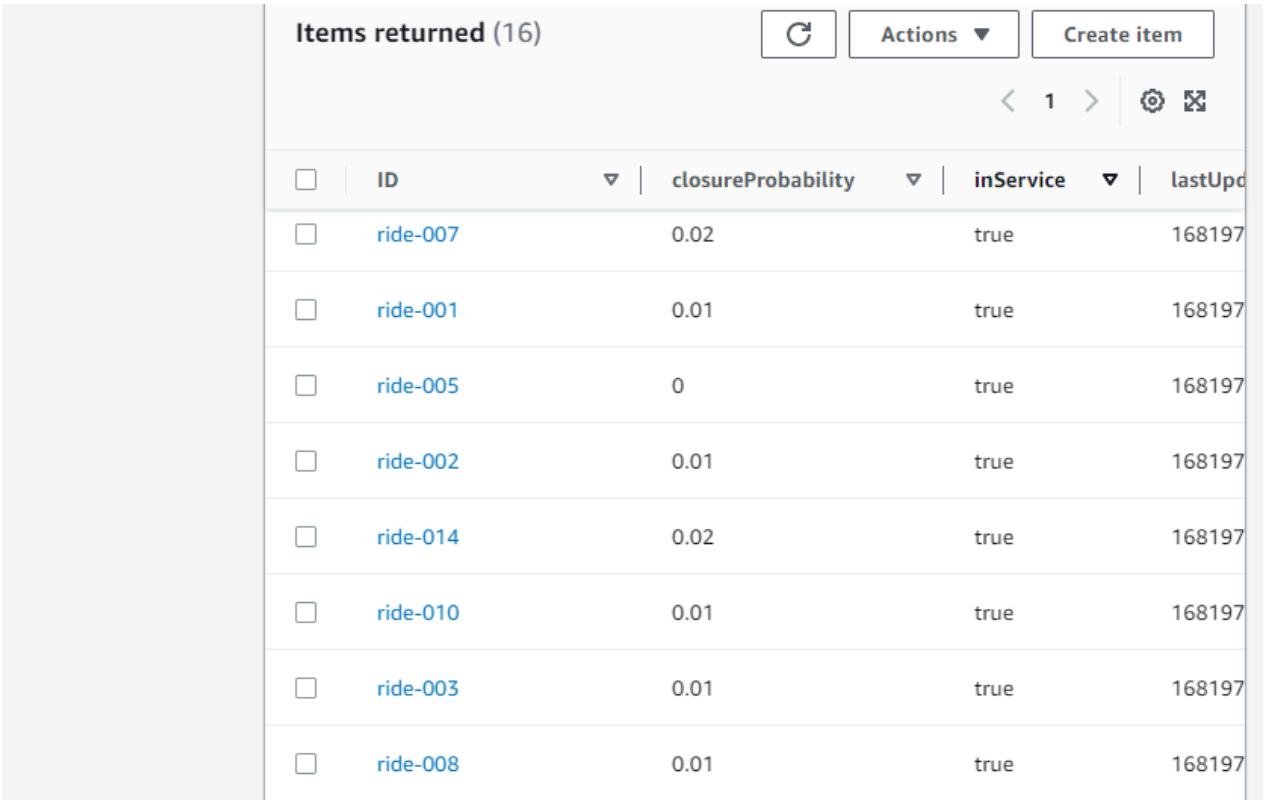
3) Responsive webpage



4) Translation



5) Table items in Dynamo DB



Items returned (16)

Actions

Create item

1

ID	closureProbability	inService	lastUpd
ride-007	0.02	true	168197
ride-001	0.01	true	168197
ride-005	0	true	168197
ride-002	0.01	true	168197
ride-014	0.02	true	168197
ride-010	0.01	true	168197
ride-003	0.01	true	168197
ride-008	0.01	true	168197

Learning outcome

To conclude, the development and deployment of a simple web application on AWS that shows a basic web page displaying information about a theme park. The project was implemented using AWS Amplify, which simplifies the process of building, testing, and deploying web applications, as well as AWS Lambda and Amazon API Gateway to create a serverless backend. Amazon DynamoDB can be used to store user data. Overall, this project demonstrates the power and flexibility of AWS services and provides a solid foundation for building and deploying more complex web applications on AWS.