Hotel Search & Booking System

Final Presentation by G1T6

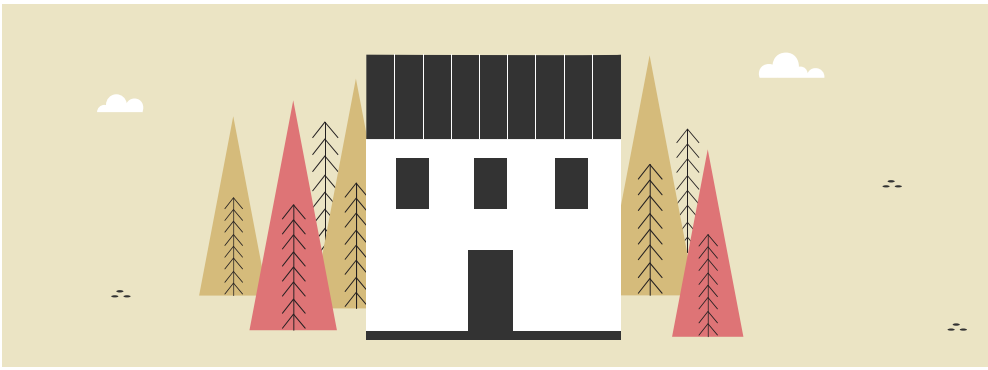Lixiang Wu (01404879)
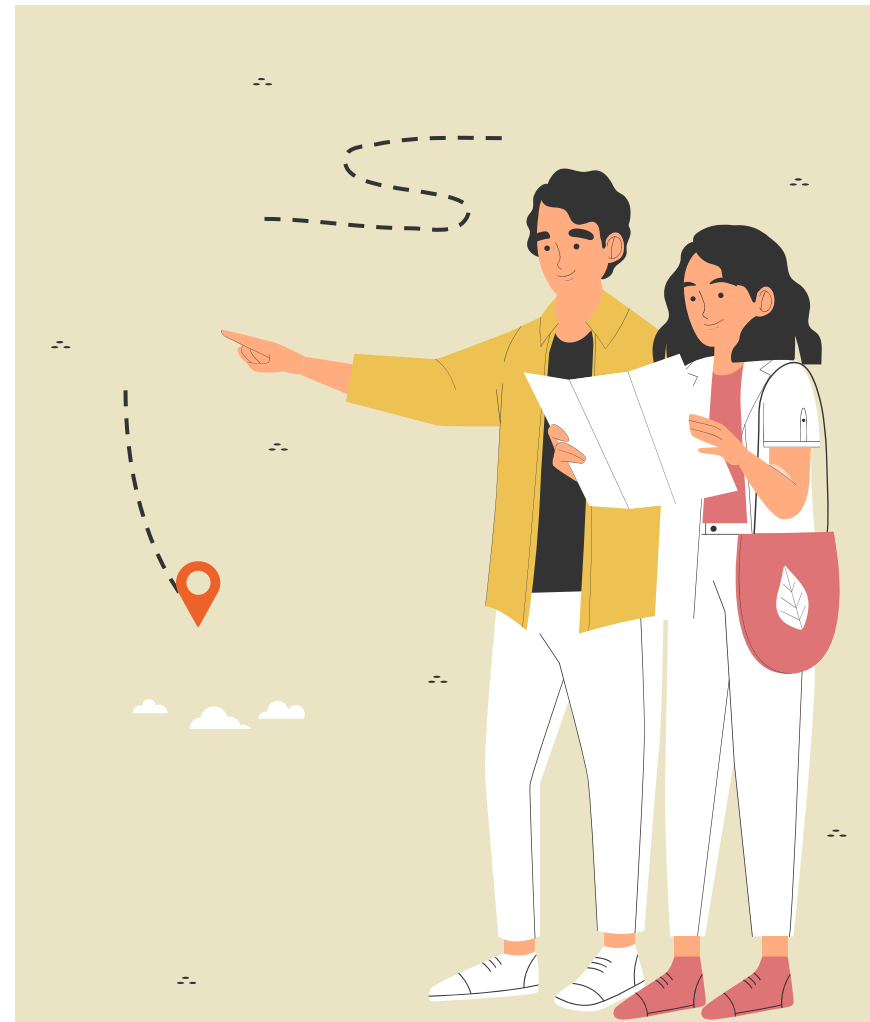Tian Mingze (01368170)
Hartono Tjakrawinata (01339300)
Soh Yu Wei (01423316)
Sia Yan Rui (01351538)
Zhang HongXiang (01366948)

Background

A Hotel Search and Booking System will be developed as requested by Ascenda Loyalty and used to power whitelabelled hotel booking platforms on behalf of banks, airlines and loyalty programs.

Stakeholders

Ascenda		Client, has master access over system and data, configurable using IAM.
Hotel Suppliers		Provide data (no special read-write access)
Users		Can view hotel rooms and rates, and can make bookings
AWS		Provide PaaS including Lambda, Aurora, etc
Developers		Provide continued maintenance, frontend and backend developers would have varied read-write system access as required.

Business requirements

Derive (some) functional requirements from business requirements

Destination Search

Hotel Search Results

Booking Data

01

02

03

Search by Destination, list provided by client

Hotel Search results returned filtered by price

Booking hotel through the application

Solution Overview
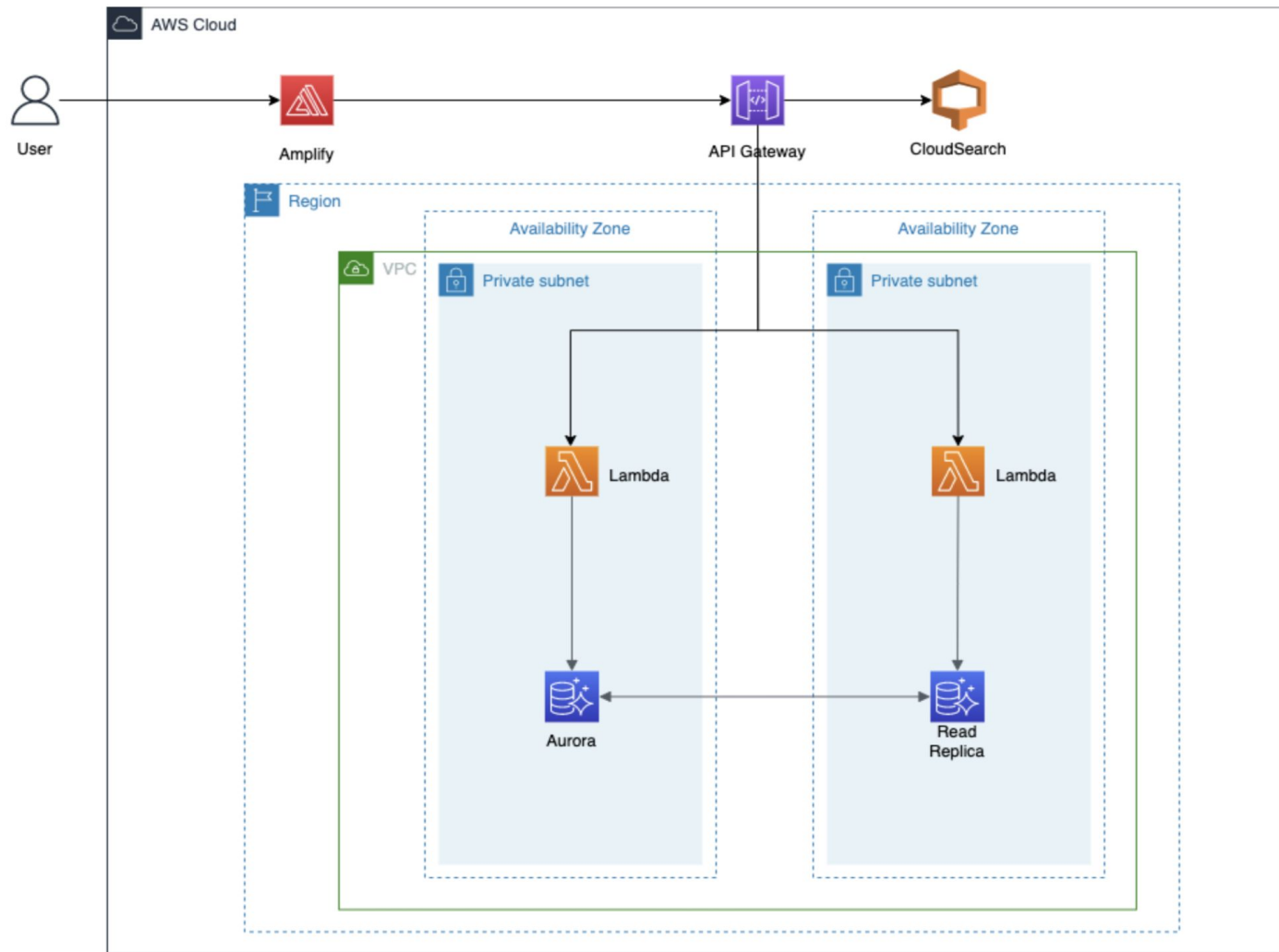
Sequence Diagram - Use Case 1
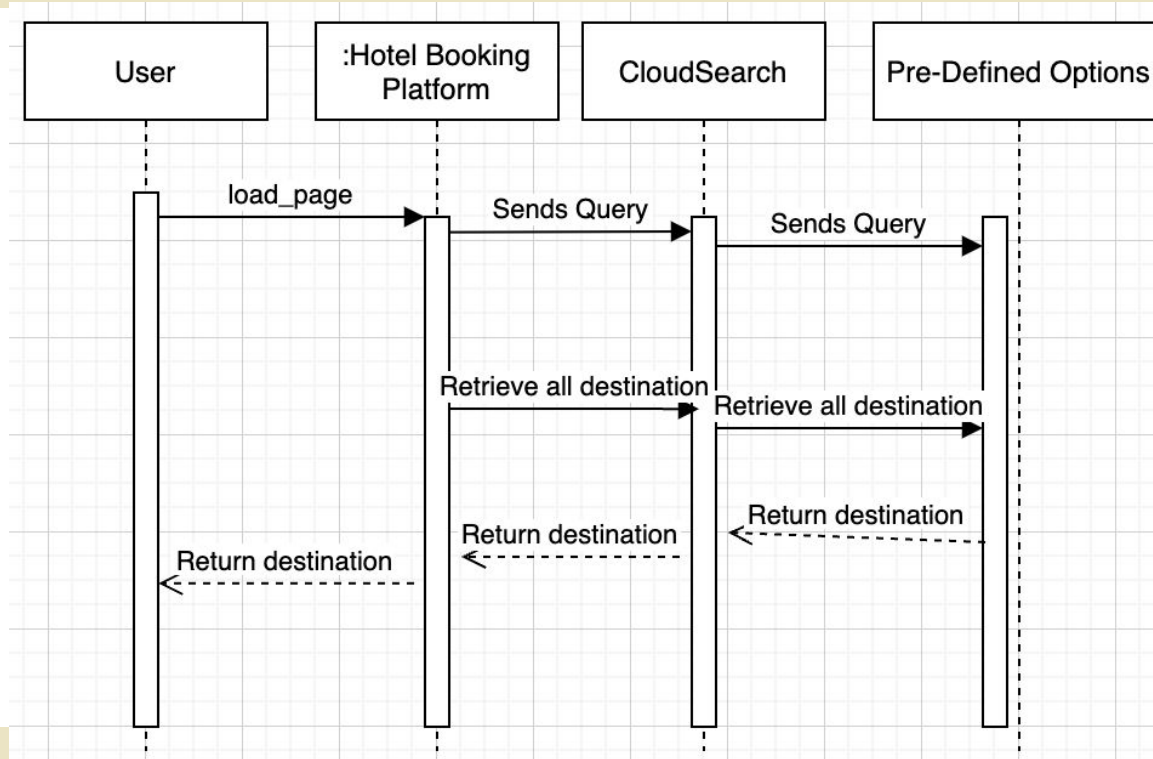
Sequence Diagram - Use Case 2
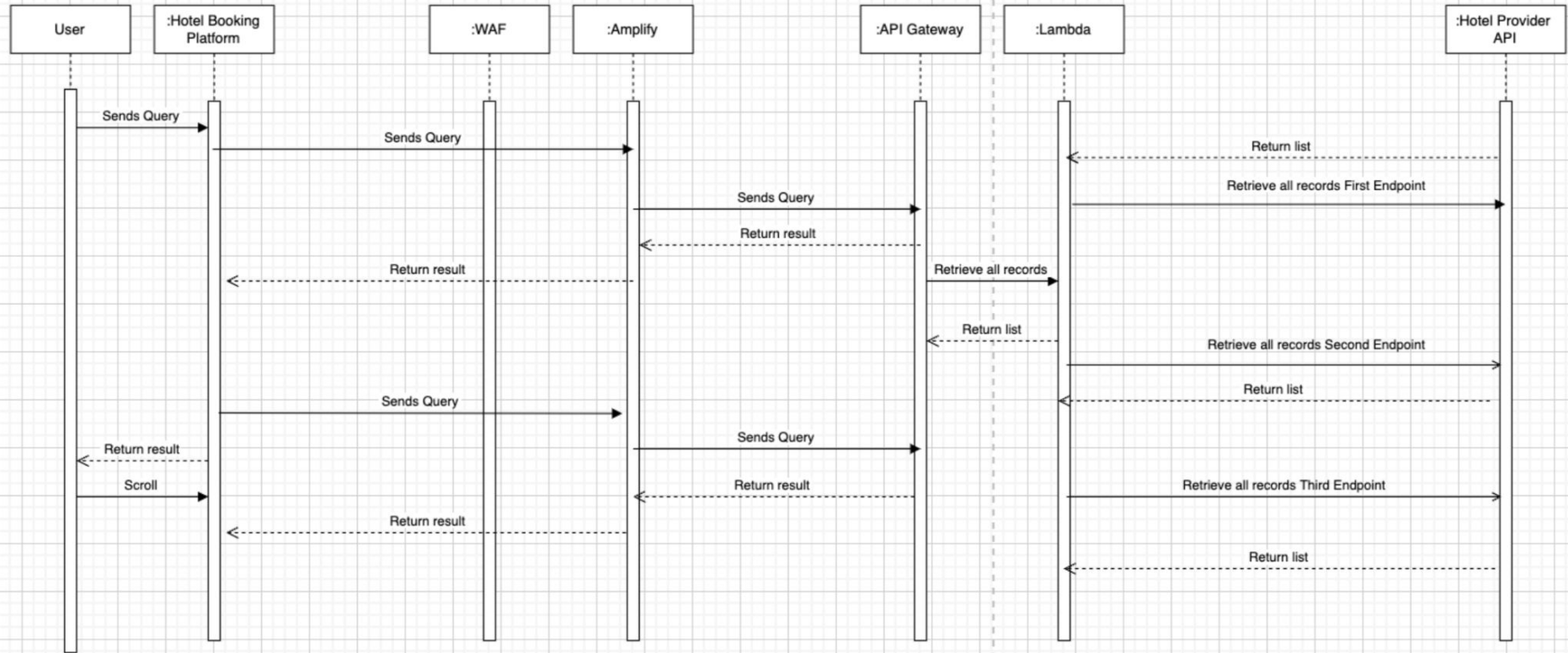
Sequence Diagram - Use Case 3
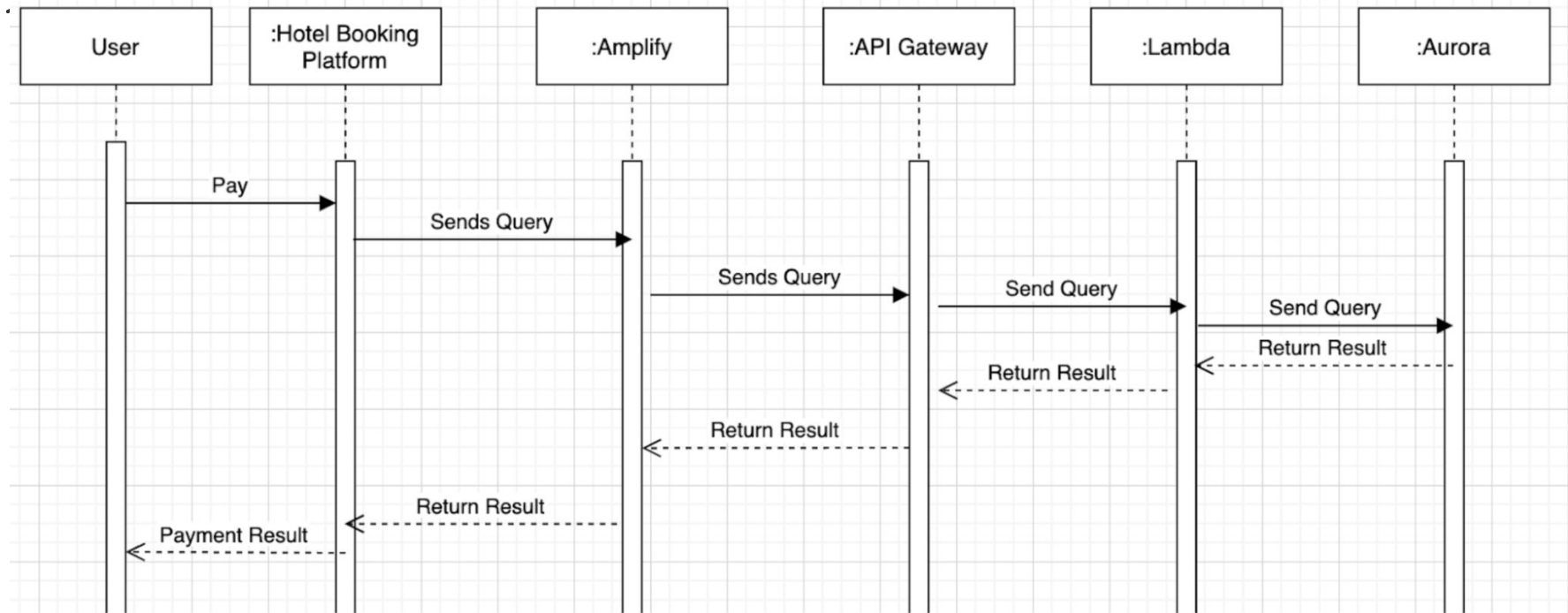
Live Demo

Hotel Search and
Booking System
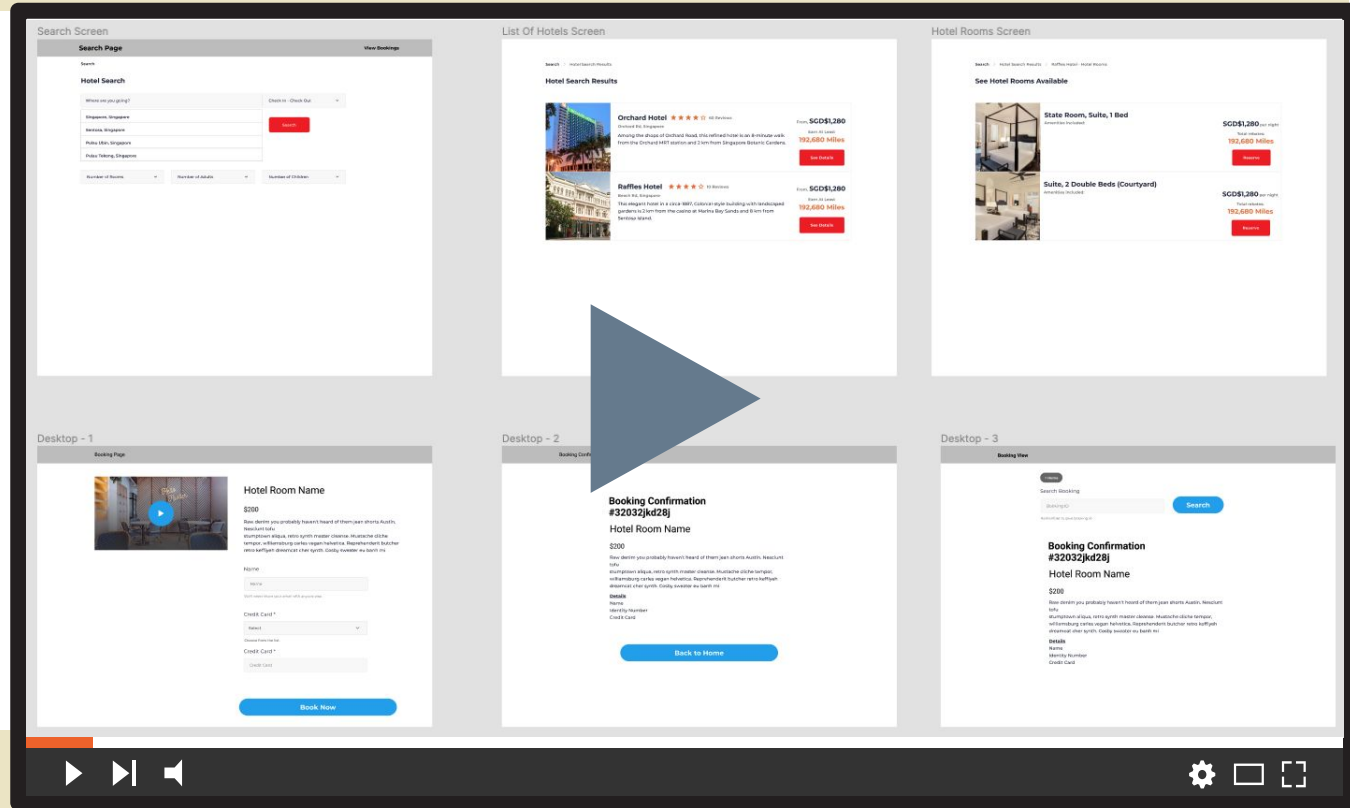
Architectural Decisions

AWS
Lambda

VS

AWS
Fargate

AWS
CloudSearch

VS

ElasticSearch
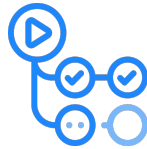
GitHub
Actions

VS

Jenkins

Architectural Decisions

AWS Aurora Serverless VS **AWS RDS**
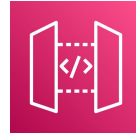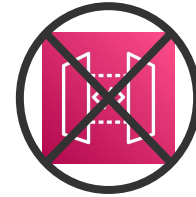
AWS API Gateway VS **No API Gateway**

Quality Requirements

Derive (some) non-functional requirements from quality requirements

Speed

Scalability

Maintainability

01

02

03

Data Security

Recovery

04

05

Quality Attribute - Speed

AWS Amplify

AWS CloudSearch

- AWS Amplify distributes the web app to users globally with low-latency and high speed transfer of data.

- AWS CloudSearch enables fast search autocomplete result with high scalability and availability.

Speed Testing

```
Summary:
  Total:        0.2304 secs
  Slowest:      0.1978 secs
  Fastest:      0.0087 secs
  Average:      0.0995 secs
  Requests/sec: 433.9421

  Total data:   6300 bytes
  Size/request: 63 bytes

Response time histogram:
  0.009 [1]   |■
  0.028 [48]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.046 [1]   |■
  0.065 [0]   |
  0.084 [0]   |
  0.103 [0]   |
  0.122 [0]   |
  0.141 [0]   |
  0.160 [0]   |
  0.179 [3]   |■■■
  0.198 [47]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

Latency distribution:
  10% in 0.0114 secs
  25% in 0.0131 secs
  50% in 0.1740 secs
  75% in 0.1855 secs
  90% in 0.1879 secs
  95% in 0.1892 secs
  99% in 0.1978 secs

Details (average, fastest, slowest):
  DNS+dialup:  0.0776 secs, 0.0087 secs, 0.1978 secs
  DNS-lookup:  0.0084 secs, 0.0000 secs, 0.0187 secs
  req write:   0.0001 secs, 0.0000 secs, 0.0035 secs
  resp wait:   0.0179 secs, 0.0084 secs, 0.0772 secs
  resp read:   0.0002 secs, 0.0000 secs, 0.0029 secs

Status code distribution:
  [400] 100 responses
```

```
Summary:
  Total:        4.7801 secs
  Slowest:      3.2872 secs
  Fastest:      0.2788 secs
  Average:      1.6597 secs
  Requests/sec: 20.9200

  Total data:   105800 bytes
  Size/request: 1058 bytes

Response time histogram:
  0.279 [1]   |■
  0.580 [46]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.880 [1]   |■
  1.181 [0]   |
  1.482 [0]   |
  1.783 [0]   |
  2.084 [2]   |■
  2.385 [0]   |
  2.686 [7]   |■■■■■■
  2.986 [28]  |■■■■■■■■■■■■■■■■■■■■■■■■
  3.287 [15]  |■■■■■■■■■■■■■

Latency distribution:
  10% in 0.3332 secs
  25% in 0.3626 secs
  50% in 2.6261 secs
  75% in 2.9133 secs
8 secs, 3.2872 secs
  DNS-lookup:  0.0116 secs, 0.0000 secs, 0.0280 secs
  req write:   0.0001 secs, 0.0000 secs, 0.0002 secs
  resp wait:   1.5609 secs, 0.2786 secs, 3.1170 secs
  resp read:   0.0002 secs, 0.0001 secs, 0.0004 secs

Status code distribution:
  [200] 100 responses
```

AWS Lambda

Amazon Aurora Serverless
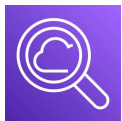
AWS CloudSearch

Quality Attribute - Scalability

- AWS Lambda, Aurora Serverless and CloudSearch are able to adjust the number of instance according to the number of events without the need for configuration or deployment due to its serverless nature.

Stress Testing

Total Requests per Second ● RPS ● Failures/s

500
400
300
200
100
0

6:09:59 pm
RPS: 430.8
Failures/s: 0
Users: 500

6:09:21 pm 6:09:28 pm 6:09:34 pm 6:09:40 pm 6:09:47 pm 6:09:53 pm 6:09:59 pm 6:10:06 pm 6:10:12 pm 6:10:19 pm 6:10:25 pm

Response Times (ms) ● Median Response Time ● 95% percentile

10,000
8,000
6,000
4,000
2,000

6:09:21 pm 6:09:28 pm 6:09:34 pm 6:09:40 pm 6:09:47 pm 6:09:53 pm 6:09:59 pm 6:10:06 pm 6:10:12 pm 6:10:19 pm 6:10:25 pm

Number of Users ● Users

600
500
400
300
200
100
0

6:09:21 pm 6:09:28 pm 6:09:34 pm 6:09:40 pm 6:09:47 pm 6:09:53 pm 6:09:59 pm 6:10:06 pm 6:10:12 pm 6:10:19 pm 6:10:25 pm

Quality Attribute - Ease of Maintenance

- AWS API Gateway allows for easy maintenance of the many APIs used within our application as it is a fully managed service by AWS

- A CI/CD pipeline automates the deployment of application changes to reduce manual errors and time spent.

Amazon API Gateway

AWS Amplify

Github

CI/CD Demo

Frontend

```
Push to Master branch in  →  Checkout repository in GitHub
the frontend folder           Actions Folder
                                      │
                                      ▼
                              Build Success?
                           No │           │ Yes
                              ▼           ▼
GitHub actions sends a email to    Push built code to AWS
GitHub user who triggered workflow  Amplify
```
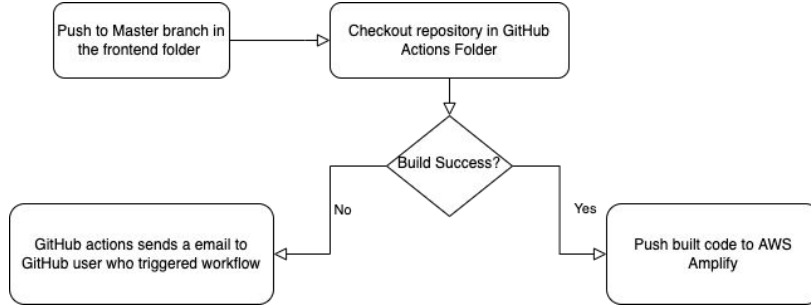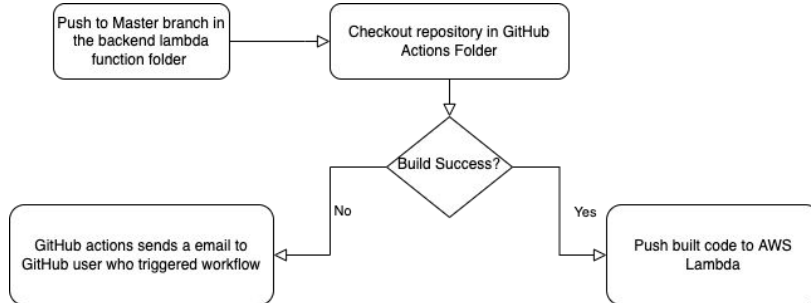
Backend

```
Push to Master branch in  →  Checkout repository in GitHub
the backend lambda            Actions Folder
function folder                       │
                                      ▼
                              Build Success?
                           No │           │ Yes
                              ▼           ▼
GitHub actions sends a email to    Push built code to AWS
GitHub user who triggered workflow  Lambda
```

Quality Attribute - Resilience & Data Recovery

Amazon API Gateway

AWS Lambda

Amazon Aurora

- AWS managed services offer several features to support data resiliency, backup needs and recovery.

- Multi-AZ architecture ensures the system remains available when unexpected events occur.

- AWS API Gateway is capable to act as a load balancer ensuring the system to be highly available.

Quality Attribute - Security

AWS Amplify

AWS IAM

AWS Secrets Manager

- AWS Amplify provides layered security perimeter against multiple types of attack.

- All data would be encrypted at rest in Amazon Aurora. Access control managed by AWS IAM and AWS Secrets Manger.

Security Test Results (OWASP ZAP)

- 📁 Alerts (7)
  - 🚩 Content Security Policy (CSP) Header Not Set (2)
  - 🚩 Missing Anti-clickjacking Header (2)
  - 🚩 Cross-Domain JavaScript Source File Inclusion (2)
  - 🚩 Timestamp Disclosure - Unix (15)
  - 🚩 X-Content-Type-Options Header Missing (6)
  - 🏳️ Information Disclosure - Suspicious Comments
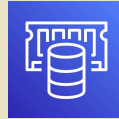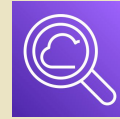 - 🏳️ Re-examine Cache-control Directives (3)

Budget

Removed:

Added:

Amazon Route 53

AWS WAF

Amazon ElastiCache

AWS CloudSearch

Budget

	Proposal	Final
Development	62.25 USD	369.02 USD
Production	86.87 USD	521.9 USD

Thanks You!