

Cloud Computing and Big Data Systems

Spring 2024

Quiz 1

- This is an open book exam. Only offline material can be used as a reference. Internet access is strictly prohibited except Brightspace.
- Set your laptop to maximum brightness and have no tab/application open other than Brightspace.
- Marks are not awarded based on the length of the answer, the more precise/concise/bullet-pointed it is, the better.

Section A - Assignments (5 x 5)

1. What protocol governs the permissions granted by a web server to web browsers when accessing resources from different domains? Explain how this works.
2. Write a Kubernetes Service manifest to expose a Deployment named "my-app" on port 8080 within the cluster.
3. Explain the concept of serverless architecture and its benefits. How does it relate to the implementation of the Dining Concierge chatbot?
4. Design a data model for the DynamoDB table "yelp-restaurants" to efficiently store and query restaurant information. Discuss the choice of partition keys, sort keys, and secondary indexes to support various query patterns, including location-based searches and cuisine filtering.
5. In a microservices architecture, when choosing between AWS EC2 and AWS Lambda for handling tasks within a service, what distinguishes their use cases?

Section B - System Design (30)

Enhanced Personalized Restaurant Recommendation Engine

Background:

Previously in your Assignment 1, you developed a Dining Concierge chatbot that provided restaurant suggestions based on user input for cuisines. However, this system did not offer personalized recommendations based on user feedback, nor did it dynamically adapt to external changes like restaurant availability or new openings.

Objective:

The goal is to develop a more intelligent and adaptive Dining Concierge system. This enhanced system should:

1. Offer personalized restaurant recommendations based on user feedback and interaction.
2. Offer recommendations based on geographic proximity.
3. Seamlessly integrate and respond to external data changes, such as updates in restaurant statuses or the introduction of new dining establishments. (assume any data source and mention it.)

So, When users request recommendations, they should receive 5 recommendations based on personal preferences, user's past searches, feedback etc and another 5 based on the trending restaurants around them.

You could assume that the application has an enhanced user screen that has a "like" button next to the name of a restaurant that is recommended to the user. You could collect these "like" as input to your trending, recommendation engine.

Requirements:

1. Data Stores

- Clearly list and describe all data stores involved and why they are a good choice.
- Define the type of data stored (use schema/ERD etc), including any indexing mechanisms, and briefly explain why this is a good design for your data store.

2. APIs

- List all new APIs that will be integrated into the system.
- Describe the low-level design for your backend briefly, focusing on AWS services and infrastructure suitable for a high-traffic system.
- Ensure that the design is scalable, event-driven, and asynchronous, capable of handling the demands of a large user base.

3. System Design Architecture (High-Level Backend Design)

- Develop an architecture diagram showing the integration of the personalized recommendation engine and dynamic data system with the existing chatbot.
- This architecture should support extensive data from user interactions and be capable of adjusting recommendations dynamically.

4. Feedback Loop, Real-Time Processing, and Adaptability Parts Working/ Explanation

- Explain the feedback loop for capturing and integrating user preferences and feedback.
- Describe how the system will handle and adapt to real-time data, including user feedback and restaurant data changes.
- Identify AWS components and services used for real-time data management.

5. Data Pipeline / Event Flow

- Detail the data pipeline (or event flow) starting from user interaction to the delivery of personalized recommendations.

This system aims to transform the Dining Concierge chatbot into a more dynamic, and user-centric service. The upgraded system should demonstrate robustness, efficiency, and the ability to adapt to the changes in the restaurant data out there and consumer preferences, leveraging AWS services effectively.

Feel free to make additional assumptions but remember to mention them.

Section C - Lectures (5 x 5)

1. Explain Master-Slave Architecture as discussed in Lecture 1 and its issues concerning database replication
2. Describe the live migration process and how it enables the seamless movement of virtual machines between physical hosts without service interruption and how it differs from traditional offline migration methods.
3. Explain how Kubernetes achieves high availability and fault tolerance in a cluster. What components and mechanisms are used to ensure that applications running in Kubernetes remain available in the event of node failures?
4. How does hybrid cloud work and what are its challenges?
5. Explain etcd, Persistent Volume, Persistent Volume Claim, and statefulsets in Kubernetes.

Section D - Research Papers (5 x 4)

1. Describe the architectural differences between Kafka and traditional enterprise messaging systems. How do these differences contribute to Kafka's efficiency and scalability in log processing?
2. Discuss the role of Chubby in maintaining the consistency and availability of Bigtable clusters, and how failures or delays in Chubby operations can affect the overall performance of Bigtable.
3. Discuss the specific challenges and trade-offs involved in implementing Dremel's nested columnar storage format compared to traditional flat relational column stores.
4. How does Borg prioritize between the need for rapid task startup and efficient resource utilization, and what trade-offs does this prioritization entail?