11/01/24 Quiz 1
6:00pm-8:15pm

- **This is an open book exam. Only PDFs and AWS Console can be used as a reference. Internet access is strictly prohibited except Brightspace and AWS Console. Handwritten / Physical Referencen notes are allowed**
- **Set your laptop to maximum brightness and have no tab/applications open other than the ones above.**
- **Marks are not awarded based on the length of the answer, the more precise/concise/bullet-pointed it is, the better.**
- **Any AI Tools, messaging applications and online note taking apps are strictly prohibited, also make sure they're not present on the dock/taskbar**

Section A: [Assignment 1 and Kubernetes] 30 points

1. You create a new Lambda function that returns the text "Hello world". You test it and it completes the execution in a few milliseconds. You then decide to use it to store a message in a DynamoDB table. You add code to do just that, on top of the original code. Upon testing the new version of the function, it times out after 3 seconds, over and over again.

   >What might be the issue and how can you resolve it?

2. When you call the API from your frontend application hosted at http://www.yourdomain.com, you get the following error in the developer console:
   "No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://www.yourdomain.com' is therefore not allowed access."

   > What is the reason for this error? How do you resolve it? (describe every step)

3. What is the purpose of a Dockerfile?
   a. To define the environment variables for a Docker container
   b. To configure the network settings for a Docker container
   c. To specify the runtime dependencies for a Docker container
   d. To define the Docker image and its dependencies

4. Which Docker component is responsible for managing images?
   a. Docker Engine
   b. Docker Registry
   c. Docker Compose
   d. Docker Hub

5. What is the main benefit of AWS Lambda?
   a. It eliminates the need for server administration
   b. It offers unlimited compute power and storage
   c. It supports only a limited number of programming languages

d. It requires manual scaling to handle increased traffic

6. How does AWS Lambda handle state management?
    a. It provides a built-in state management system
    b. It does not handle state management, and requires the use of external storage systems
    c. It uses an in-memory cache for state management
    d. It relies on the programmer to handle state management within the function code

7. A _____ breakdown fields values of a document into a stream, and inverted indexes are created and updated using these values, and these stream of values are stored in the document.
    a. Analyzer
    b. Shard
    c. Filter
    d. Tokenizer

8. Which of the following is a collection of fields in a specific manner defined in JSON format?
    a. Node
    b. Shard
    c. Index
    d. Document

9. Which of the following runs on each node and ensures containers are running in a pod?
    a. Pod
    b. Etcd
    c. Kubelet
    d. Scheduler

10. Which field in a Service YAML file specifies which pods the service should route traffic to?
    a. selector
    b. ports
    c. type
    d. targetPort

Section B: System Design[30]

Restaurant Recommendation Engine with ***Current Wait-time and Special Offers (if any)***

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 with current wait-time in the respective restaurant.
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.)

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. These recommendations should include the current wait-time and if there is any special offer from the restaurant.

For wait-time, you assume that restaurants have an API that when you query, will provide you the current wait-time estimate. For the special offers, you assume that there is an API the restaurant provides what is the special offer for the next day if any. The offer expires the next day.

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 with the wait-time and offers information.

4. Feedback Loop, Real-Time Processing, and Adaptability Parts Working/ Explanation
   - Explain the feedback loop for capturing and integrating user preferences and feedback.
   - Explain how you will handle the current wait-time and special offers APIs and how you would use these in a scalable way.
   - 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: [Lecture Notes] 20

1. How does the trade-off between memory space and false positives influence the design of Bloom filters in large databases, and how might this impact overall database performance?
2. Explain the key difference between Full Virtualization, and Para Virtualization and state the pros-cons of each.
3. Imagine you are developing a containerized application where user-uploaded files need to be persistently stored across container restarts. How would you use Docker volumes to achieve this, and provide a small Docker Compose template illustrating this.
4. You built a docker image where the dockerfile looks like follows:
   Docker image 1 —----------
      1) COPY ./Assignment ./src
      2) RUN cd ./src
      3) RUN “some build operation” —----> point 1

   Docker image 2 —----------
      1) COPY ./Assignment ./src
      2) RUN cd ./src
      3) RUN “some build operation”

4) RUN “rm -rf ./src”
5) RUN “ls” —----------> point 2

What do you think about the image size of docker image 1 and docker image 2, which one will be greater ? Explain your answer

Section D: [Papers] 20

1. Describe the concept of eventual consistency in DynamoDB and how it supports high availability.
2. In a large-scale MapReduce job with multiple node failures, describe how MapReduce’s fault tolerance mechanisms would ensure task completion without data loss.
3. How do the design architectures of the Google File System (GFS) and Bigtable differ in terms of data storage, scalability, and fault tolerance, and what are the implications of these differences for their performance and suitability for various applications?
4. In the Borg system, what role does the priority system play in ensuring resource allocation for high-priority tasks, especially during resource contention?