Fall23 Quiz1 answers

Section A: Based on Assignment 1 working knowledge [25 Points]

1. In assignment 1, you created an API Gateway with a POST method. You also created a Lambda function that can handle the POST request. What are some reasons for using API Gateway instead of directly invoking the Lambda function?

1. Reasons for using API Gateway instead of directly invoking the Lambda function:

- API Gateway provides additional features like request/response transformation, caching, and throttling[8].
- It offers better security through API keys and authorization methods[8].
- API Gateway can handle CORS, which is important for browser-based applications[8][14].
- It provides a unified interface for multiple backend services, not just Lambda functions[8].

2. In assignment 1, what is a session in Lex? How can it be useful?

2. A session in Lex:

- A session in Lex represents the state of a conversation between a user and the bot[13].
- It's useful for maintaining context across multiple interactions, storing user preferences, and managing the flow of the conversation[13].

3. In assignment 1, you notice that some of the documents are not being indexed in your elasticsearch index, and there are no error messages. How would you troubleshoot this scenario and ensure that all documents are indexed correctly? List at least 2 points of failures.

3. Troubleshooting documents not being indexed in Elasticsearch:

- Check Elasticsearch logs for any error messages or rejected requests[9].

- Verify that the indexing process is not paused due to settings or configuration issues[9].

- Ensure that the Elasticsearch cluster has enough resources to handle the indexing load[9].

- Check for mapping issues or incompatible data types in the documents being indexed[9].

4. In assignment 1, When developing a RESTful API with AWS API Gateway, you encounter an issue where certain HTTP requests, particularly those with non-standard headers, are failing unexpectedly. You suspect that preflight requests might be involved. Explain the concept of preflight requests, why they are important for web security, and how you would troubleshoot and resolve this issue in your API Gateway setup

4. Preflight requests, their importance, and troubleshooting in API Gateway:

- Preflight requests are OPTIONS requests sent by browsers before making certain cross-origin requests to check if the actual request is safe to send[5][14].

- They are important for web security as they help prevent unauthorized cross-origin requests[5].

- To troubleshoot, ensure CORS is properly configured in API Gateway, including allowed origins, methods, and headers[14].

- For ANY routes, create an unauthorized OPTIONS route with a mock integration to handle preflight requests[14].

5**. In assignment 1, you created a DynamoDB table for storing restaurant data. You used restaurant/business ID as the primary key. What could possibly go wrong if you use insertedTimestamp as the primary key? How about using insertedTimestamp as the secondary key**

5. Issues with using insertedTimestamp as the primary or secondary key in DynamoDB:

- As a primary key: It could lead to hot partitions if many items are inserted at the same time, causing uneven distribution of data and potential throttling[15].

- As a secondary key: It may not provide efficient querying patterns if the primary access pattern is not time-based, and could still lead to uneven distribution of data across partitions[15].

Section B: System Design [30 Points] You want to extend the Assignment 1 to add the following features:

1. Based on user preferences and past search, 5 restaurants are recommended to the user via email or SMS everyday.

2. Whenauser searches for restaurants, the daily specials from those restaurants, if any, are also shown to the user. Note that these deals are released by restaurants on a daily basis through an imaginary Yelp API, say dailySpecial that you are allowed to invoke. You need to extend the architecture to accommodate the above two features. You need to state all the required data model details, APIs that need to be supported. You need to draw the end-to-end event interaction for the above two features to illustrate your idea details.

1. For recommending 5 restaurants daily via email or SMS:

- Use Amazon DynamoDB to store user preferences and search history.

- Implement an AWS Lambda function to run daily, querying user data and generating recommendations.

- Use Amazon SES (Simple Email Service) or Amazon SNS (Simple Notification Service) to send recommendations via email or SMS.

2. For showing daily specials when searching for restaurants:

- Create an API to fetch daily specials from the imaginary Yelp API.

- Modify the existing search Lambda function to include a call to this new API.

- Update the DynamoDB schema to include a field for daily specials.

- Modify the search results to include the daily special information.

Data model details and API support would need to be designed based on specific requirements and existing implementation details.

Section C: Papers [25 points]

1. GFS: Using Fig 1, describe how a chuck server failure is handled in GFS

1. GFS chunk server failure handling:

- The master node detects the failure through regular heartbeat messages[16].
- It then updates its metadata to reflect the unavailable chunks[16].
- Replication is used to ensure data availability, so other replicas can serve the data[16].

2. GFS: Why single GFS master is used? How does GFS recover from GFS master failure?

2. Single GFS master usage and recovery:

- A single master is used for simplicity and to maintain global knowledge of the filesystem[16].
- Recovery from master failure involves using a replicated transaction log and checkpoints[16].
- Backup masters can replay the log to recover the state quickly[16].

3. Dremel architecture design:

(No specific information provided in the search results about Dremel's architecture.)

4. Two key reasons for Dremel's speed:

(No specific information provided in the search results about Dremel's speed.)

5. BigQuery: What are key differences between BigQuery and MapReduce? Give examples of use cases where BigQuery is preferred over MapReduce and vice versa

5. Key differences between BigQuery and MapReduce:

- BigQuery is designed for interactive data analysis, while MapReduce is for batch processing of large datasets[10].

- BigQuery uses SQL for querying structured data, while MapReduce requires custom programming for data processing[10].

- BigQuery is better suited for ad-hoc querying and BI, while MapReduce is better for complex data processing and unstructured data[10].

Section D: Lecture Notes [20 Points]

1. Why is hybrid cloud beneficial for an enterprise compared to either public or private cloud

- Combines the security of private cloud with the power and flexibility of public cloud[11].

- Allows for better control over sensitive data and compliance requirements[11].

- Provides cost-effectiveness by optimizing resource allocation between on-premises and cloud environments[11].

- Enables easier scaling and growth compared to purely on-premises solutions[11].

2. What is paravirtualization and why is paravirtualization more efficient than full virtualization?

2. Paravirtualization and its efficiency:

- Paravirtualization is a virtualization technique where the guest OS is modified to work with specific APIs instead of hardware[7].

- It's more efficient than full virtualization because it reduces the overhead of emulating hardware[7].

- The guest OS can communicate directly with the hypervisor, leading to better performance[7].

3. Explain how iterative memory copy is leveraged for live migration

3. Iterative memory copy for live migration:

- Iterative memory copy involves transferring the VM's memory content to the target host in multiple rounds[6].

- It starts by copying the entire memory, then in subsequent iterations, only the pages that were modified (dirty pages) during the previous round are copied[6].

- This process continues until the amount of remaining dirty pages is small enough to allow for a brief pause and final transfer, minimizing downtime[6].

4. What are the key differences between VM and containers

4. Key differences between VM and containers:

- VMs virtualize the entire computer system including hardware, while containers virtualize only the operating system[3].

- VMs run a full OS for each instance, while containers share the host OS kernel[3].

- VMs are larger in size and take longer to start, while containers are lightweight and start quickly[3].

- VMs provide stronger isolation and security, while containers offer higher density and efficiency[3].