kris-classes /
restart-q4-2023-assignment2-amy

Code Issues Pull requests Actions Projects Security Insights

main

restart-q4-2023-assignment2-amy / **assignment-2-v1.md**

mechatroNick Initial commit 2 months ago

Rules

- Its ok to discuss strategy/solution between each other, just don't copy work
- Don't make it obvious that you have copied unmodified contents from the AI tools/Stackoverflow
- Don't ask the tutors to give you the answer, though you may ask for hints/high level how

- Feel free to ask the tutors to get you fix an execution time error/environment setup fault
- Feel free let us know during class if you find something is wrong with the assignment questions
- Be reasonable with the tutors' time, so everyone can have a turn at asking them for help

Part I. System architecture and system design (30 marks total)

These are the research based questions, in short-answer questions form. Each question is expecting 150-300 words for answers for each sub part (I.1., I.2....). Attachment of images, and reference links are encouraged.

Please use `Part I` folder to contain your answers in `markdown files`

References:

- Serverless land
- Enterprise Integration Pattern
- System design primer

Part I.1. (5/30 marks)

- Explain eventual and strong consistency
- Which AWS persistent services/feature should you expect eventual consistency?
- Which AWS persistent services/feature should you expect strong consistency?
- Name some (2-3) usecases where eventually consistent persistences are acceptable

Answer in file `Part I/Part I.1.md`

Part I.2. (5/30 marks)

- Explain the need for messaging integration
- Why should we use message brokers in microservices architecture?
- Find AWS architecture references for the use of message broker in microservices.

Answer in file `Part I/Part I.2.md`

Part I.3. (5/30 marks)

- Explain the differences between point-to-point messaging pattern (Queue) and publish/subscribe pattern (Pub/Sub)

- When would you use point-to-point messaging over the other?
- When would you use publish/subscribe over the other?
- Find AWS architecture references for point-to-point messaging
- Find AWS architecture references for publish/subscribe

Answer in file `Part I/Part I.3.md`

Part I.4. (5/30 marks)

- Explain idempotency in the context of transactional processing
- What does it mean to have an idempotent consumer?
- Given SQS as the message broker, and Lambda function as the message consumer
  - How would you implement Lambda as an idempotent consumer?
  - Which component would become the bottleneck when you implement idempotent consumers?

Answer in file `Part I/Part I.4.md`

Part I.5. (5/30 marks)

- What is Change Data Capture (CDC)?
- What is CDC useful for?
- Which additional architecture patterns can CDC enable?
- Find AWS architecture references for the use of CDC
  - Transactional Processing
  - Analytics Processing

Answer in file `Part I/Part I.5.md`

Part I.6. (5/30 marks)

- Given a simple AWS EC2 in a Auto Scaling Group architecture with ALB and a dedicated DB instance, name 5 system design components/techniques that can help you scale your system to meet more end-customers demand. Hinted layers:
  - Static content cache
  - Dynamic content cache
  - Compute distribution
  - Persistency
  - Decoupling methods
- For each component, explain in few words (other than adding complexity), what are the trade offs of introducing these components if they weren't added before

Answer in file `Part I/Part I.6.md`