

[Skip to content](#)

based on both the sample quizzes attached - YOU are supposed to generate sample questions **FOR SECTION A** that align with the material of assignment1 (we dont have assgn 2)

refer the following docs:

1. Assignment1_Task_Sheet
2. Assgn1_Github_Link --> <https://github.com/rohan-g0re/AWS-Dining-Concierge-Chatbot>

- Get in detail with AWS services used in the assignment
- REFER THE ARCHITECTURE DIAGRAM in the task sheet
- Go through each and every task in the assignment (total 7) --> identify nitty gritties f how you were supposed to implement this using AWS Console (refer @aws docs)

Based on all of this --> DRAFT QUESTIONS WHICH MATCH THE LEVEL OF section a questions in both the quizzes --> ALSO add a few question that are TOUGHER than the questions in samples

THINK HARD and draft LOTS OF QUESTIONS

Thought for 40s >

Section A — Assignment-1-Aligned Sample Questions (AWS Dining Concierge)

Below is a large bank of **Section A-style** questions (mostly MCQs + a few short answers) that mirror the difficulty and tone of the provided quizzes, with several **harder ★ items** mixed in. They're organized by the **7 tasks** and the **architecture** in your A1 spec.

Task 1 — Frontend on S3 (static hosting)

Reference: A1 §1 + S3 website hosting link. [Assignment1_Task_Sheet](#)

1. Which S3 setting must be enabled to serve `index.html` publicly as a static site?

- a) Transfer Acceleration
 - b) Static website hosting
 - c) Requester Pays
 - d) Intelligent-Tiering
2. You uploaded files to `myapp-bucket` but get **403 Forbidden** in browser. What's the most likely missing step?
- a) Enable versioning
 - b) Add a bucket policy allowing `s3:GetObject` on `arn:aws:s3:::myapp-bucket/*`
 - c) Turn on Glacier Deep Archive
 - d) Create a lifecycle rule
3. For S3 static sites, what is the **website endpoint** domain pattern?
- a) `https://s3.amazonaws.com/<bucket>`
 - b) `http://<bucket>.s3-website-<region>.amazonaws.com`
 - c) `https://<bucket>.cloudfront.net`
 - d) `https://<bucket>.s3.<region>.amazonaws.com`
4. You host the starter UI and need your frontend to call your API Gateway with signed requests later. Which change is **not** required on S3?
- a) Enable public access block
 - b) Upload `index.html`, `app.js`
 - c) Set index & error documents
 - d) (Optional) Add CloudFront for HTTPS/custom domain
5. Short answer: Name **two** reasons S3 works well for a frontend in A1 (durability/availability/scale).

Lecture_3_Microservices

Task 2 — API Gateway + Lambda (LF0) + CORS

Reference: A1 §2 (Swagger import, CORS, SDK).

[Assignment1_Task_Sheet](#)

[Assignment1_Task_Sheet](#)

6. You imported the provided Swagger into API Gateway. What does the import primarily create?
- a) A Lambda layer
 - b) REST resources, methods, models, and integrations as defined
 - c) A DynamoDB table
 - d) An SQS queue
7. The browser shows **CORS** errors when calling your API. Which **two headers** are essential in the API method response for preflight?
- a) `Access-Control-Allow-Origin`, `Access-Control-Allow-Methods`
 - b) `Content-Encoding`, `ETag`
 - c) `X-Amz-Date`, `Authorization`
 - d) `Retry-After`, `Cache-Control`

8. Where is the **easiest place** to skip to content for all methods on a resource in API Gateway?
- a) In Lambda console
 - b) In API Gateway “Enable CORS” on the resource/method
 - c) In S3 bucket properties
 - d) In DynamoDB global tables
9. Your LF0 returns a JSON body, but the frontend receives a stringified JSON. What mapping fix in API Gateway usually resolves this?
- a) Add `integration.response.body` mapping template to `application/json`
 - b) Enable proxy integration and let Lambda return proper JSON with headers
 - c) Set usage plan quota
 - d) Add VPC link
10. MCQ: Which deployment step is **required** for clients to see new API changes?
- a) Create a new stage or re-deploy to an existing stage
 - b) Rotate IAM role
 - c) Increase Lambda memory
 - d) Rebuild S3 site
11. ★ Your team wants the **generated JavaScript SDK** to sign requests automatically. Which **combo** provides this?
- a) API Gateway SDK + Cognito Identity (temporary IAM creds) + SigV4
 - b) API Gateway SDK + EC2 metadata
 - c) CloudFront signed cookies
 - d) S3 pre-signed URLs only
- (Your lectures emphasize SDK generation + SigV4 with Cognito for signed calls.)

[Lecture_3_Microservices](#)

Task 3 — Lex Bot + Lambda Code Hook (LF1) + SQS (Q1)

Reference: A1 §3 (intents, slots, code hook, push to SQS).

[Assignment1_Task_Sheet](#)

[Assignment1_Task_Sheet](#)

12. In Lex, which component validates slots **before** Lex responds?
- a) Slot type
 - b) Alias
 - c) Lambda code hook
 - d) Utterance list
13. Which **three base intents** are required by A1?
- a) Greeting, Help, Cancel
 - b) Greeting, ThankYou, DiningSuggestions
 - c) Launch, BookTable, Goodbye

RO

- d) Welcome, Confirm, Retry [Skip to content](#)

Assignment1_Task_Sheet

14. The DiningSuggestionsIntent must capture which fields at minimum? (Select **all** that apply)
- a) Location
 - b) Cuisine
 - c) Dining Time
 - d) Number of people
 - e) Email
 - f) Phone number

(Answer includes a-e per spec.) [Assignment1_Task_Sheet](#)

15. After collecting slots, where must you **enqueue** the user request?
- a) SNS topic
 - b) SQS queue (Q1)
 - c) Kinesis stream
 - d) EventBridge bus

Assignment1_Task_Sheet

16. MCQ: Right after pushing to Q1, what must the bot send to the user?
- a) SES email with results
 - b) A confirmation that suggestions will be emailed shortly
 - c) A pre-signed S3 link
 - d) A payment link

Assignment1_Task_Sheet

17. You see Lex defaulting to **FallbackIntent** even for clear phrases. Which could be true?
- a) Not enough sample utterances
 - b) Bot not **built/published** to an alias
 - c) Code hook errors
 - d) Any of the above

(Modeled after sample quiz style.) [Sample_Quiz_Spring2025](#)

18. Short answer: In your code hook, name **two** common validation checks on slots (e.g., time in future, valid cuisine list).
19. MCQ: For SQS Q1, which property defines how many times a failing message is retried before going to DLQ (extra credit)?
- a) VisibilityTimeout
 - b) MessageRetentionPeriod
 - c) maxReceiveCount on redrive policy
 - d) DelaySeconds
- (Extra credit DLQ described explicitly.) [Assignment1_Task_Sheet](#)
20. ⭐ Your LF1 sometimes times out when calling SQS. Which **two** mitigations help first?

- a) Increase Lambda timeout Skip to content
 - b) Use SendMessageBatch to ... Assignment1_Task_Sheet
 - c) Add provisioned concurrency
 - d) Switch SQS to FIFO automatically
-

Task 4 — Chat API calls Lex (wire Lex into LF0)

Reference: A1 §4 (send user text to Lex, return its response). Assignment1_Task_Sheet

21. Put the steps in order for LF0 when a chat message arrives:
 - i) Send text to Lex
 - ii) Return Lex result to client
 - iii) Extract message text from API body
 - iv) Await Lex response(Correct order: iii → i → iv → ii.) Assignment1_Task_Sheet
 22. MCQ: If LF0 runs in a VPC without NAT, Lex calls may fail due to...
 - a) Lambda cold start
 - b) No internet egress to reach Lex endpoint
 - c) IAM role missing lex:PutBot
 - d) Missing CloudWatch alarms
 23. Short answer: Name **one** reason to keep the Lex bot decoupled from email sending logic. (Hint: async worker later.)
-

Task 5 — Yelp data ingestion + DynamoDB (yelp-restaurants)

Reference: A1 §5 (5 cuisines × ~200 each, dedupe, required attributes, insertedAtTimestamp). Assignment1_Task_Sheet Assignment1_Task_Sheet

24. How many **minimum cuisines** and **approx. count per cuisine** are expected?
 - a) 3 cuisines × 100
 - b) 5 cuisines × ~200
 - c) 10 cuisines × 50
 - d) 2 cuisines × 500Assignment1_Task_Sheet
25. Which table name is mandated?
 - a) restaurants
 - b) yelp-restaurants
 - c) manhattan-yelp
 - d) dining-dbAssignment1_Task_Sheet
26. Which attribute is **explicitly required** when storing each item?

- a) updatedAtTimestamp
- b) insertedAtTimestamp
- c) randomScore
- d) cuisineGroup

[Skip to content](#)[Assignment1_Task_Sheet](#)

27. Choose the **required fields** to retain for recommendations (select all):

- a) Business ID
- b) Name
- c) Address
- d) Coordinates
- e) Number of Reviews
- f) Rating
- g) Zip Code

(All of the above a–g are listed.) [Assignment1_Task_Sheet](#)

28. Short answer: List **two** common dedupe strategies across multiple Yelp queries (e.g., by business id; by phone+zip).

29. MCQ: Where may you legally run the scraping job?

- a) Only from EC2
- b) From your laptop or AWS—your pick
- c) Only from Lambda
- d) Only from on-prem VM

[Assignment1_Task_Sheet](#)

30. ★ Suppose you hit Yelp API rate limits. Which mitigation **does not** help?

- a) Distributed backoff per cuisine
- b) Cache/business-id seen set
- c) Randomize offsets
- d) Hard-retry in tight loop

31. DynamoDB primary key choice most aligned with lookups later is:

- a) PK= restaurant_id
- b) PK= cuisine , SK= restaurant_id
- c) PK= zip , SK= rating
- d) PK= name , SK= address

(Hint: later we fetch by IDs gathered from search.)

Task 6 — OpenSearch/Elasticsearch index for cuisine → restaurant IDs

Reference: A1 §6 (index/type and fields to store). [Assignment1_Task_Sheet](#)

32. Which **index** and **type** does A1 require?

- a) food/restaurants

- b) Index restaurants , type Skip to content
- c) dining/nyc
- d) Any default index

33. What **minimal fields** must each ES doc store?

- a) RestaurantID and Cuisine
- b) Name , Address , Rating
- c) Zip , Coordinates
- d) ReviewText , Stars

Assignment1_Task_Sheet

34. You need **random** suggestions for a cuisine. What's a simple approach without heavy scoring?

- a) function_score with random seed
- b) terms aggregation only
- c) script_fields with Math.random() and client-side shuffle
- d) Index sorting by _id

35. MCQ: To restrict results to a **single cuisine**, you should use:

- a) Range query
- b) Match-all query
- c) Term filter on cuisine
- d) Geo distance filter

36. Your ES domain is **VPC-only** and LF2 times out. The fix is most likely:

- a) Increase Lambda memory only
- b) Put LF2 in same VPC/subnets + security group access to ES
- c) Create API key
- d) Switch to S3

Task 7 — Suggestions Worker (LF2) + SES + EventBridge + DLQ

Reference: A1 §7 (pull SQS, query ES + DynamoDB, email via SES, schedule every minute). Assignment1_Task_Sheet

37. Which service should **invoke LF2 periodically**?

- a) CloudWatch Events/EventBridge Scheduler
- b) CloudTrail
- c) Step Functions
- d) SNS

Assignment1_Task_Sheet

38. Which **lookup order** matches A1?

- a) DynamoDB → ES → SES
- b) ES (get IDs by cuisine) → DynamoDB (hydrate details) → SES (email)
- c) SES → ES → DynamoDB

- d) ES only

[Skip to content](#)

Assignment1_Task_Sheet

39. SES in sandbox requires which action before sending to an address?

- a) Verify sender domain only
- b) Verify **recipient** email (or move out of sandbox)
- c) Request KMS key
- d) Upload DKIM private key

40. When LF2 **fails to send email**, A1 extra credit expects:

- a) Delete SQS message immediately
- b) Leave message; let SQS retry; then move to DLQ after `maxReceiveCount`
- c) Push to SNS
- d) Rebuild the API

Assignment1_Task_Sheet

41. Short answer: Name **two** items you would log to CloudWatch when email sending fails (e.g., `requestId`, `error`, `message body keys`). Assignment1_Task_Sheet

42. ★ Your EventBridge rule is set to `rate(1 minute)` but LF2 **stops** pulling messages. Which is **least** likely?

- a) LF2 IAM lacks `sqs:ReceiveMessage`
- b) SQS policy denies Lambda principal
- c) ES index name changed
- d) Lambda concurrency set to 0 (disabled via reserved concurrency)

Architecture & Cross-Cutting (A1 diagram & lectures)

Architecture: see **ANNEX** in A1; lectures reinforce microservices + serverless stack

(API GW + Lambda + Cognito + S3 + SQS + SES).

Assignment1_Task_Sheet

Lecture_3_Microservices

Lecture_3_Microservices

43. Which path best describes the **end-to-end** user flow in A1?

- a) Frontend (S3) → API GW → LF0 → Lex → (LF1 validates + enqueues) → Q1 → LF2 → ES+DDB → SES
- b) Frontend → EC2 → RDS → SES
- c) Frontend → S3 → CloudFront → SES
- d) Lambda → EKS → RDS

44. What is the key **decoupling point** that separates chat from emailing?

- a) S3 bucket
- b) SQS queue Q1
- c) CloudTrail
- d) Kinesis

45. MCQ: Which AWS service combination in A1 most directly implements a **serverless microservices** style?

- a) EC2 Auto Scaling + ALB Skip to content
- b) API Gateway + Lambda + Cognito + S3 + DynamoDB + (OpenSearch)
- c) EKS + StatefulSets
- d) EMR + HDFS

(Your lecture explicitly highlights API GW + Lambda + Cognito + S3 + SQS + SES as building blocks.) [Lecture_3_Microservices](#) [Lecture_3_Microservices](#)

46. ★ Why is an **async** pattern (queue + worker) promoted for workflows like suggestions? Pick the **least** valid reason.

- a) Absorb traffic spikes and third-party slowness
- b) Reduce user-visible latency
- c) Completely eliminate need for error handling
- d) Improve resilience to partial failures

(Lectures emphasize async for spikes/failures.) [Lecture_3_Microservices](#)

47. From a security standpoint for signed API calls from web, which trio is highlighted?

- a) IAM + Access Keys in JS bundle + Secrets Manager
- b) Cognito Identity + temporary IAM creds + SigV4 SDK
- c) KMS + S3 + ACM
- d) EC2 instance profile only

[Lecture_3_Microservices](#)

48. Short answer: State **two** advantages of hosting the frontend on S3 for A1 (hint: durability & availability).

[Lecture_3_Microservices](#)

49. ★ (Tough) You want to avoid **CORS preflights** from your S3 site to API GW. Name **one** technique that reduces preflights without breaking security (e.g., use only GET, simple headers; or route via same-origin domain with CloudFront + custom domain).

50. Your repo link (A1 codebase) to cite in your write-up: **what is the GitHub URL?**

(Short answer.) [Assgn1_Github_Link](#)

“Code/CLI Mini” (A-style quick wins)

51. Write a one-liner AWS CLI to sync `./dist/` to an S3 bucket used for your static site (erase removed files).

- *Short answer expected.* (Modeled after 2023 quiz's CLI snippets.) [Sample_Quiz_Spring2023](#)

52. Given A1 requires `insertedAtTimestamp`, show a JSON **PutItem** fragment for DynamoDB that sets an ISO timestamp for that attribute. (Short answer; pseudo/JSON acceptable.) [Assignment1_Task_Sheet](#)

53. Show a **minimal** SQS `SendMessage` call in a Lambda (language-agnostic pseudocode) that includes cuisine, email, and location. (Short answer.)

54. ★ Provide an example **ES query** that returns random restaurants for a given cuisine (e.g., `function_score` with `random_score`). (Short answer.)

55. Write the **EventBridge schedule expression** to trigger LF2 every minute. (Short answer.) [Assignment1_Task_Sheet](#)

[Skip to content](#)

“Ops & Limits” (quiz-style questions)

56. A sudden surge causes **immediate API failures** in a serverless chat API. The most likely cause is:

- a) Lambda **concurrency** limit reached
- b) Low memory
- c) Lex slots exhausted
- d) Usage plan missing

(Modeled after sample.) Sample_Quiz_Spring2025

57. After a period of inactivity, your first request to LF0 is slow. The most likely cause is:

- a) Cold start
- b) Lex throttling
- c) ES shard reallocation
- d) DDB WCU exceeded

(Modeled after sample.) Sample_Quiz_Spring2025

58. ★ Email sending works locally but fails in AWS with SES sandbox errors. What's the **first** check?

- a) Verify recipient or move account out of sandbox
- b) Create VPC endpoint for SES
- c) Increase Lambda memory
- d) Switch to SES SMTP

59. Your LF2 logs show repeated retries of the **same** SQS message but never success.

What setting dictates when it lands in DLQ?

- a) `VisibilityTimeout`
- b) Redrive policy `maxReceiveCount`
- c) Message retention
- d) Delivery delay

Assignment1_Task_Sheet

60. (Tough) You decide to **add Cognito** later to sign frontend API calls. Name one **benefit** highlighted in lectures (e.g., temporary IAM creds + SigV4 signing via SDK).

Lecture_3_Microservices

If you want, I can turn this into a **printable quiz sheet** with an answer key on a separate page.



just give me the answer key with simple explanations

Thought for 29s >

[Skip to content](#)

Section A — Answer Key (with quick explanations)

- 1) b. You must enable **Static website hosting** to serve `index.html`.
- 2) b. Public reads need a **bucket policy** allowing `s3:GetObject` on `bucket/*`.
- 3) b. S3 website endpoint is `http://<bucket>.s3-website-<region>.amazonaws.com`.
- 4) a. Enabling **Block Public Access** would prevent public website access; not required.
- 5) **Durable, highly available** (and cheap/ininitely scalable) \Rightarrow perfect for static frontends.
- 6) b. Swagger import creates **resources/methods/models/integrations** in API Gateway.
- 7) a. Preflight needs `Access-Control-Allow-Origin` + `Access-Control-Allow-Methods` (often `-Headers` too).
- 8) b. Use API Gateway's **Enable CORS** on the resource/method.
- 9) b. **Lambda proxy integration** returns proper JSON + headers without mapping templates.
- 10) a. **Deploy** (to a stage) for clients to see API changes.
- 11) a. API GW **SDK + Cognito Identity** (temp IAM creds) = automatic **SigV4** signing.
- 12) c. **Lambda code hook** validates slots before Lex responds.
- 13) b. **Greeting, ThankYou, DiningSuggestions**.
- 14) a,b,c,d,e. Location, Cuisine, Time, Party size, **Email** (phone not mandated).
- 15) b. Enqueue to **SQS** (Q1).
- 16) b. Reply that **suggestions will be emailed shortly**.
- 17) d. Any of the above (few utterances, not built/published, code hook errors).
- 18) Examples: **time in future**, cuisine \in allowed list, **valid email**, party size in range, Manhattan-only.
- 19) c. **maxReceiveCount** in the redrive policy controls DLQ handoff.
- 20) a & b. Increase Lambda **timeout**; use **SendMessageBatch** to cut calls.
- 21) iii \rightarrow i \rightarrow iv \rightarrow ii. Extract \rightarrow call Lex \rightarrow await \rightarrow return.
- 22) b. No **NAT** = no internet egress to reach Lex endpoint.
- 23) Asynchronicity: **keep chat fast**; email done later by worker; **isolates failures**.
- 24) b. 5 cuisines \times ~200 each.
- 25) b. Table name: `yelp-restaurants`.
- 26) b. Must include `insertedAtTimestamp`.
- 27) a-g (all). ID, Name, Address, Coords, Reviews, Rating, Zip.
- 28) E.g., dedupe by **Yelp business_id**; or **phone+zip** set; maintain a **seen-ID set**.
- 29) b. From laptop or AWS—your choice.
- 30) d. **Tight-loop hard retries** worsen rate limits; use backoff/caching instead.
- 31) a. PK = **restaurant_id** (fast GetItem after ES returns IDs).

- 32) b.** Index restaurants , type Skip to content
- 33) a.** Minimal: **RestaurantID** and **cuisine**.
- 34) a.** function_score with **random_score** for random picks.
- 35) c.** term filter on cuisine .
- 36) b.** Put Lambda in **same VPC/subnets/SG** to reach VPC-only ES.
- 37) a.** EventBridge/CloudWatch Events scheduled trigger.
- 38) b.** ES → DynamoDB → SES (IDs by cuisine → hydrate → email).
- 39) b.** In SES **sandbox**, verify recipient (or move out of sandbox).
- 40) b.** Let SQS retry; DLQ after **maxReceiveCount**.
- 41)** Log **requestId/error + message keys** (cuisine, email, location, IDs).
- 42) c.** ES index rename wouldn't stop the **schedule**; IAM/SQS/conc. issues would.
- 43) a.** S3 → API GW → LF0 → Lex → (LF1 validate+enqueue) → SQS → LF2 → ES+DDB → SES.
- 44) b.** **SQS Q1** is the decoupling point.
- 45) b.** API GW + Lambda + **SQS + SES + DDB + OpenSearch** = serverless microservices.
- 46) c.** Async does **not** eliminate error handling.
- 47) b.** Cognito Identity + temp IAM + SigV4 from SDK.
- 48)** Durability & availability (plus low cost, scale) for static content.
- 49)** Use **simple requests** (GET, simple headers) to avoid preflights **or** front with CloudFront + same-origin custom domain/proxy.
- 50)** <https://github.com/rohan-g0re/AWS-Dining-Concierge-Chatbot>
- 51)** aws s3 sync ./dist s3://<your-bucket> --delete
- 52)** Example PutItem attrs:

json

Copy code

```
{
  "restaurant_id": {"S": "abc123"},
  "insertedAtTimestamp": {"S": "2025-10-22T22:15:00Z"}
}
```

- 53)** Pseudocode:

pseudo

Copy code

```
sqS.sendMessage({
  QueueUrl: Q1,
  MessageBody: JSON.stringify({ cuisine, email, location, time, partySize })
})
```

- 54)** Random ES query:

json

[Skip to content](#) [Copy code](#)

```
{
  "query": {
    "function_score": {
      "query": { "term": { "cuisine": "thai" } },
      "random_score": {}
    }
  },
  "size": 5
}
```

55) EventBridge: `rate(1 minute)`

56) a. Lambda concurrency hit → throttling/429s.

57) a. Cold start after idle period.

58) a. Verify recipient or get out of **sandbox**.

59) b. DLQ handoff controlled by **maxReceiveCount**.

60) Temporary IAM creds + auto SigV4 signing via SDK; no long-lived keys in browser.

...

Can you explain me following questions in detail:
9, 19, 20, 22, 28, 36, 42, 46, 49, 58, 59, 60

Thought for 33s >

Here you go—each one unpacked with what's happening, why it happens, how to fix it, and a couple of gotchas.

Q9 — API Gateway returns “stringified JSON” to the frontend

1. What's going on

- In **non-proxy** integrations, API Gateway needs **mapping templates** to transform the Lambda result into a proper HTTP response. If the template is missing/misaligned, your JSON can get double-encoded (i.e., frontend sees a quoted JSON string).

2. The clean fix

- **Enable Lambda proxy integration** for that method. Then have your Lambda return:

json

 [Copy code](#)

{

[Skip to content](#)

```

    "statusCode": 200,
    "headers": { "Content-Type": "application/json" },
    "body": "{\"ok\":true}" // stringified JSON
}

```

- With proxy integration, API Gateway forwards status/headers/body **as is**; the browser sees Content-Type: application/json and your `fetch(...).json()` works as expected.
3. Other acceptable fix (if staying non-proxy)
 - Add an **integration response mapping template** for application/json that sets the body to the raw JSON (not a quoted string).
-

Q19 — How SQS hands messages to a DLQ (redrive)

1. Key setting

- The **redrive policy's** `maxReceiveCount` on the *source* queue determines how many times a message can be received (and fail to be processed) before SQS moves it to the **dead-letter queue**.

2. How the count increments

- Each time the message becomes **visible again** (after the visibility timeout expires without a successful delete), and is **received** again, the receive count increments.

3. Interplay with visibility timeout & Lambda

- If your Lambda doesn't `DeleteMessage` (or fails), the message reappears after **VisibilityTimeout** and will be retried until `maxReceiveCount` is hit → then it's sent to the DLQ for offline triage.
-

Q20 — LF1 sometimes times out calling SQS

1. First two levers (the answers you picked)

- **Increase Lambda timeout** (gives the SDK time to retry/transmit).
- Use `SendMessageBatch` to reduce per-message round-trips and overhead.

2. Extra reliability tips

- **Connection reuse**: keep the SQS client outside the handler for connection pooling.
 - **Exponential backoff + jitter** on retries.
 - If your Lambda is in a **private subnet**, add a **NAT Gateway** or an **SQS VPC endpoint** (Gateway Endpoint) so it can reach SQS reliably.
-

Q22 — Lambda in a VPC can't reach Lex

1. Root cause
 - **Amazon Lex is a public service.** A Lambda placed in **private subnets without NAT** has **no outbound internet** and can't reach Lex endpoints.
 2. Fix options
 - Add a **NAT Gateway** to those private subnets (standard pattern).
 - Or **don't place** this Lambda in a private subnet if it doesn't need VPC resources.
 - (There is no VPC endpoint for Lex; you need internet egress.)
-

Q28 — Dedupe strategy for Yelp ingestion

1. Why dupes happen
 - Multiple queries/offsets, overlapping cuisine searches, or pagination can return the **same business** multiple times.
 2. Practical dedupe keys
 - **Primary:** Yelp `business_id` (stable and unique).
 - **Secondary** (fallback): **phone + postal code** or **(normalized name + address)** to catch odd cases.
 3. How to implement
 - Maintain an **in-memory** `seen_ids` **set** during scraping.
 - Enforce uniqueness at storage (e.g., DynamoDB PK on `restaurant_id`).
 - Log counts so you can confirm ~200 unique per cuisine.
-

Q36 — Lambda cannot reach a VPC-only OpenSearch/Elasticsearch domain

1. What “VPC-only domain” means
 - Your OpenSearch domain has **no public endpoint**; it's reachable **only inside the VPC** via VPC networking + security groups.
2. Required setup
 - Put your **Lambda in the same VPC/subnets** as the domain (typically **private subnets**).
 - Configure **security groups**: allow **inbound 443** on the domain SG **from the Lambda's SG**; Lambda SG should allow **egress** to the domain SG.
 - Ensure **route tables/NACLs** permit traffic between those subnets.
 - Confirm the domain's **resource policy / fine-grained access** allows the Lambda's role.
3. Quick test
 - From the function (or a throwaway EC2 in the same SG/subnets), `curl https://<domain-endpoint>/_cluster/health`.

[Skip to content](#)

Q42 — EventBridge triggers but LF2 “stops pulling messages”

1. What the options imply
 - **IAM lacks sqs:ReceiveMessage**: Lambda is invoked but can't pull → no processing.
 - **SQS policy denies Lambda principal**: same effect—invoked but blocked at the queue.
 - **Reserved concurrency set to 0**: Event fires, but Lambda is effectively **disabled**.
 - **ES index name changed**: LF2 still runs and pulls SQS; it just fails later when querying ES.
 2. Therefore
 - The **least likely** root cause of “stops pulling” is the **ES index rename**—that would cause processing errors, not prevent pulling.
-

Q46 — Async (queue + worker) does *not* remove error handling

1. Real benefits of async
 - **Absorb spikes**, isolate slow/fragile dependencies, and keep the **user response fast**.
 - **Retry** transient failures without blocking the user.
 2. What you still must do
 - **Idempotency** (so retries don't duplicate side effects).
 - **DLQs and poison-pill handling** (bad payloads go somewhere safe).
 - **Visibility timeout tuning**, partial-batch failure reporting, and **observability** (logs/metrics/traces).
 3. Bottom line
 - Async improves resilience and latency, but **you still need robust error handling**.
-

Q49 — Reducing or avoiding CORS preflights

1. Reduce preflights with “simple requests”
 - Use **GET/POST** only.
 - Restrict headers to **simple headers** (no custom `x-*`, no `Authorization`).
 - Use **Content-Type** one of: `text/plain`, `application/x-www-form-urlencoded`, or `multipart/form-data`.
2. Avoid CORS altogether via same-origin

- Put your site and API url → **gin** (same scheme+host+port). Skip to content.
- Common pattern: CloudFront → Lambda behaviors—
 - / → S3 website
 - /api/* → API Gateway (with a **custom domain** on the same host)
- From the browser's POV, both paths share the **same origin**, so **no CORS**.

3. Notes

- If you must send credentials or custom headers, you'll trigger preflights—then configure CORS properly on API Gateway.
-

Q58 — SES sandbox: why email “works locally” but fails in AWS

1. SES sandbox rules

- In **sandbox**, you can *only* send from verified identities **to** verified identities (or after your account is out of sandbox).
- Local success ≠ AWS success if local tests used different recipients/senders.

2. What to do

- **Verify the recipient email** (and sender).
 - Or request **production access** (move account out of sandbox) in SES console
→ Account details / Sending limits.
-

Q59 — Exactly when SQS moves a message to DLQ

1. The mechanism

- The **source queue** has a **redrive policy** pointing to a DLQ and a `maxReceiveCount`.
- Every time a message is **received** and then **not deleted** before visibility timeout, its **receive count increments**.

2. The moment of handoff

- Once the count **exceeds** `maxReceiveCount`, SQS **moves the message to the DLQ automatically**.
 - Lambda batch quirks: if the function fails the **whole batch**, all messages re-appear and their counts may all increment together; use **partial batch item failures** to only retry bad items.
-

Q60 — Why use Cognito Identity + SigV4 from the browser

1. Core benefits

- **No hard-coded keys** in Skip to content
 - Users receive **temporary, ~~long-term~~ credentials** via Cognito **Identity Pools** (federation or unauthenticated identities).
 - The generated SDK (or AWS SDK) handles **SigV4 signing** automatically.
2. Security/wrangling perks
- Easy **key rotation** (STS short-lived creds).
 - **Least-privilege** policies per identity (even per group/role).
 - Cleaner story for **audit** and **revocation** vs. shipping static keys in JS.
-

If you want, I can turn these into flashcards (question on one side, “why/how” on the other) for quick midterm drills.



the question set of 60 questions looked pretty easy and the options "gave it out" about answers

genereate a tougher question set of 20 questions where the questions follow difficulty ofthe sample quizzes and the options are tough to = NOT TRICKY OPTIONS - BUT NOT BLANTANTLY UNDERSTANDABLE OPTIONS

Section A — Tougher 20-Question Set (Assignment-1 Focus)

MCQs unless noted. Some are “choose 2” or “choose 3” — it’s stated when applicable.

1. Your S3 static site loads `index.html` but fetches to API Gateway fail with `CORS` errors even after enabling CORS on individual methods. Which **single** change most reliably fixes *all* resources under `/prod` without per-method edits?
 - a) Add `Access-Control-Expose-Headers` on S3
 - b) Re-deploy the API stage *after* using **Enable CORS** at the **resource level** and confirm `OPTIONS` is created
 - c) Increase API Gateway timeout to 60s
 - d) Turn off S3 Block Public Access
2. You imported the Swagger/OpenAPI and switched the method to **Lambda proxy integration**. The browser still shows a quoted JSON string in the response. What’s

- the **most likely** cause? [Skip to content](#)
- a) Lambda returned body as `... ,..., ...` instead of a string
 - b) Missing Content-Type: application/json header in Lambda's return
 - c) API stage cache not invalidated
 - d) Binary media types not configured
3. Your Lex code hook validates slots and enqueues to SQS. Occasionally the **same email** receives two suggestion emails for one request. What's the **best** mitigation?
- a) Increase SQS message retention
 - b) Add an **idempotency key** (e.g., Lex inputTranscript hash) and dedupe before send
 - c) Lower EventBridge rate
 - d) Increase Lambda memory for LF2
4. You need **random** cuisine suggestions from OpenSearch **without bias** toward recently indexed docs. Which approach is the **most appropriate** for stable randomness per request?
- a) query_string with wildcard + client-side shuffle
 - b) function_score with random_score and a **varying seed** per request
 - c) Sort by _id and pick top N
 - d) Use terms aggregation on RestaurantID
5. LF2 runs in a VPC; the OpenSearch domain is **VPC-only**. LF2 sometimes times out, but only during traffic spikes. Which **pair** addresses the **network path** and **burst handling** together? (choose 2)
- a) Put LF2 in the **same subnets/SGs** as the domain
 - b) Enable Lambda **provisioned concurrency**
 - c) Add a NAT Gateway
 - d) Increase OpenSearch EBS volume size
6. You want the frontend to avoid preflights **and** keep a single origin. What is the **cleanest** setup?
- a) Route 53 → A record to S3 website endpoint
 - b) **CloudFront** distribution with two behaviors: / → S3, /api/* → API Gateway **custom domain** on the same host
 - c) S3 static website + API Gateway default execute-api domain
 - d) API Gateway **private** endpoint with VPC Link
7. Lambda LF1 occasionally exceeds its timeout when sending to SQS. CloudWatch shows multiple TCP connects. What's the **single** code change that most helps before scaling out?
- a) Instantiate the SQS client **outside** the handler for connection reuse
 - b) Switch to FIFO queue
 - c) Reduce function memory
 - d) Use short polling
8. Your SES is in **sandbox**. Users report "Email not received" for `foo@example.com`. Sender identity is verified. Which step is **required** to deliver?

- a) Verify recipient `foo@example.com` Skip to content first production access
b) Allocate dedicated IPs
c) Enable DKIM only
d) Raise account concurrency
9. You need to fetch **restaurant details by ID** in DynamoDB after OpenSearch returns IDs. Which key design gives the most direct, scalable reads?
- a) PK= `restaurant_id` (no sort key)
b) PK= `cuisine`, SK= `restaurant_id`
c) PK= `zip`, SK= `rating`
d) PK= `name`, SK= `address`
10. For the Lex → SQS path, you want to **minimize duplicates** caused by retries. Which **two** controls are most effective together? (choose 2)
- a) Short SQS visibility timeout
b) **Idempotency token** stored with message and checked by LF2
c) SQS **long polling** for LF2
d) Strict FIFO with content-based deduplication across all cuisines
11. Your API sometimes returns **429** to the S3 frontend during a class demo. CloudWatch shows Lambda **throttles**. What's the most targeted, short-term remediation while keeping cost reasonable?
- a) Raise Lambda **reserved concurrency** for the API's function
b) Add WAF rate limiting on API Gateway
c) Increase function memory from 128MB to 1GB
d) Enable X-Ray
12. EventBridge triggers LF2 every minute. SQS has growing `ApproximateNumberOfMessagesVisible`. Lambda metrics show **no invocations** during the last 10 minutes. What's the **first** console check?
- a) EventBridge rule target set to the wrong alias/version of LF2
b) OpenSearch domain health green/yellow/red
c) SQS redrive policy
d) DynamoDB throttling
13. You enable API Gateway **usage plans** and API keys as a guardrail for classmates testing your API. Which behavior is **correct**?
- a) API keys authenticate the caller with IAM
b) API keys are **not auth**; they gate access to a **usage plan** and quotas
c) API keys are required for Lambda proxy
d) API keys remove the need for CORS
14. A teammate enables S3 “**Block Public Access: On**” on the website bucket and also attaches a bucket policy allowing `s3:GetObject` to * on `bucket/*`. The site stops loading. Why?
- a) Bucket policy lost priority to SCP
b) Block Public Access **overrides** the public bucket policy denies/permits

- c) Need versioning for webs' Skip to content
d) Need Requester Pays off
15. LF2 processes SQS messages in **batches of 10**. One item is malformed and throws.
Which behavior is **most accurate** with default settings?
a) Only the bad item is retried
b) All 10 messages become visible again after visibility timeout
c) The batch is auto-split and partial success is recorded
d) The bad item is deleted, others succeed
16. You want to **lower perceived latency** of the first Lex interaction from the S3 site after idle periods. Which approach is **most relevant**?
a) Increase API Gateway timeout above 29s
b) **Provisioned concurrency** on the **API Lambda (LF0)**
c) Increase DynamoDB WCU
d) Raise SQS message retention
17. You plan to add **Cognito Identity Pools** so the generated API SDK signs requests from the browser. Which statement is **correct**?
a) Identity Pools issue **temporary IAM creds** that the SDK uses for SigV4
b) Identity Pools require verified emails in SES
c) User Pools are required for temporary IAM
d) Identity Pools remove the need for IAM roles
18. OpenSearch queries sometimes fail from LF2 with `403` while the domain is reachable. You already placed LF2 in the same VPC and SG. What's the **next** most probable cause?
a) Missing OpenSearch **resource policy/FGAC** permission for the Lambda **IAM role**
b) NAT Gateway missing
c) Route 53 alias not set
d) Lambda memory too low
19. You want to **cut API preflights** without CloudFront. Which **two** changes reduce preflights while preserving security? (*choose 2*)
a) Use only **simple headers** and `Content-Type: application/x-www-form-urlencoded`
b) Add custom `x-Requested-With` header
c) Switch all calls to `PUT`
d) Keep methods to **GET/POST** where possible
20. SQS → DLQ flow seems inconsistent in tests. Sometimes messages take longer to reach DLQ. Which pair best explains the variance? (*choose 2*)
a) **Visibility timeout** length and whether the function deletes the message
b) The `maxReceiveCount` on the redrive policy
c) S3 bucket replication status
d) EventBridge rule target ARN format



[Skip to content](#)

for each question, EXPLAIN --> question, intuition behind making a choice, reason for choosing the answer, answer

the intuition should not directly be answer - it should tell how OUT OF ALL THE OPTIONS we can figure out to look into the direction of the answer

Thought for 26s >

Here you go—each item has: **Question → Intuition → Reason → Answer.**

1) CORS still failing after enabling per-method CORS

Question. What single change fixes CORS across all resources under /prod without editing every method?

Intuition. With API Gateway, CORS changes aren't live until **deployed to a stage**; also,

📁 > ChatGPT 5 Thinking ▾

Share

...

Reason. Using **Enable CORS** on the **resource** (not just method), then **re-deploying** the stage ensures generated `OPTIONS` responses and headers are active for all child methods under that resource.

Answer. b) Re-deploy after **Enable CORS** at the resource level and confirm `OPTIONS` was created.

2) Still seeing a “quoted JSON string” with Lambda proxy integration

Question. After switching to **Lambda proxy**, the browser still shows quoted JSON.

Intuition. In proxy mode, API GW passes back what Lambda returns. If the browser shows “stringified JSON” as plain text, headers are often missing/mis-set.

Reason. You must set headers: `{"Content-Type": "application/json"}` in the Lambda response. Without it, clients/DevTools may treat the payload as plain text.

Answer. b) Missing `Content-Type: application/json` header in Lambda's return.

3) Duplicate suggestion emails for a single request

Question. What's the best mitigation for accidental duplicate sends?

Intuition. Duplicates happen with systems like Redis. The canonical antidote in distributed systems is **idempotency**.

Skip to content

Reason. Include an **idempotency key** (e.g., hash of user input / Lex request ID) in the message and dedupe at LF2 (cache/store “already processed” keys).

Answer. b) Add an idempotency key and dedupe before send.

4) Random suggestions from OpenSearch without bias

Question. How to get random cuisine results, not skewed to recency?

Intuition. True server-side randomness per request comes from query-time scoring, not from client shuffle or `_id` ordering.

Reason. `function_score` with `random_score` and a **varying seed** produces unbiased randomness each request.

Answer. b) `function_score + random_score` with per-request seed.

5) LF2 → VPC-only OpenSearch times out during spikes (choose 2)

Question. Which pair fixes both **network path** and **burst** handling?

Intuition. First, ensure the **path** exists (same VPC/subnets/SGs). Then smooth cold-start/burst pressure with **provisioned concurrency**.

Reason. (a) guarantees connectivity; (b) reduces cold starts under burst. NAT is irrelevant for VPC-only domain access; EBS size doesn’t fix timeouts.

Answer. a) + b).

6) Avoid preflights and keep a single origin

Question. What’s the cleanest setup to keep same-origin and ditch CORS?

Intuition. Same host/scheme/port = same origin. The standard way to multiplex S3 site and API under one host is **CloudFront** with multiple behaviors.

Reason. Put `/` → S3, and `/api/*` → API GW **custom domain** on the same hostname via CloudFront → same origin, no CORS.

Answer. b) CloudFront with two behaviors; API GW custom domain on same host.

7) LF1 timeouts to SQS; m? Skip to content cts in logs

Question. What single code change helps most before scaling?

Intuition. Excess TCP connects = no connection reuse. AWS SDK clients are expensive to re-create on every invoke.

Reason. Instantiate the SQS client **outside** the handler so it's reused across invocations in the same execution environment.

Answer. a) Create SQS client outside the handler.

8) SES sandbox: recipients aren't receiving emails

Question. Sender is verified; what is required to deliver to `foo@example.com` ?

Intuition. In **sandbox**, both **sender and recipient** must be verified (or you must exit sandbox).

Reason. Verify the recipient (or request production access). DKIM, IPs, and concurrency aren't the blockers here.

Answer. a) Verify recipient or move account out of sandbox.

9) Best DynamoDB key design for “fetch by ID”

Question. After OpenSearch returns IDs, what key schema gives direct reads?

Intuition. You'll be doing `GetItem` by **restaurant_id** repeatedly; a simple PK fits.

Reason. `PK = restaurant_id` gives O(1) point reads without secondary lookups or scans.

Answer. a) `PK= restaurant_id`.

10) Minimize Lex→SQS→LF2 duplicates (choose 2)

Question. Which two controls most effectively reduce duplicate sends?

Intuition. Prevent **processing** duplicates (idempotency) and **queue-level** duplicates (FIFO/content-based dedupe) where applicable.

Reason. (b) Idempotency prevents duplicate side effects; (d) FIFO with content-based dedupe avoids duplicate enqueues of the same payload within the de-dupe window.

Answer. b) and d).

[Skip to content](#)

11) 429s during demo; CloudWatch shows Lambda throttles

Question. What's the targeted, cost-sane short-term fix?

Intuition. Throttling = hitting concurrency caps. The narrowest fix is raising **reserved concurrency** for that function.

Reason. It increases available concurrent executions for the API Lambda without broad account-wide changes.

Answer. a) Raise Lambda **reserved concurrency**.

12) EventBridge every minute, SQS backlog up, zero LF2 invocations

Question. First console check?

Intuition. If invocations are zero, the trigger likely isn't invoking the right target (wrong alias/ARN) or is disabled.

Reason. Verify the **EventBridge rule target** points to the correct **LF2 alias/version** (and the rule is enabled).

Answer. a) Check rule target (alias/version) in EventBridge.

13) API Gateway usage plans + API keys behavior

Question. Which statement is correct?

Intuition. API keys are **not authentication**; they meter and gate access (quotas/usage plans).

Reason. Usage plans apply quotas/limits; keys don't replace IAM/Cognito and don't affect CORS.

Answer. b) Keys gate usage plans/quotas, not auth.

14) S3 “Block Public Access: On” + public bucket policy

Question. Site stops loading—why?

Intuition. Global **Block Public Access** overrides public ACLs/policies at the bucket level.

Reason. With BPA on, the public `GetObject` policy is ignored → 403s.

Answer. b) BPA overrides the public bucket policy.

[Skip to content](#)

15) Batch size 10; 1 item throws error. What happens by default?

Question. What happens by default?

Intuition. Without **partial batch failure** enabled, Lambda treats the **whole batch** as failed.

Reason. All 10 messages re-appear after visibility timeout; enable partial batch item failure to acknowledge good ones only.

Answer. b) All 10 reappear after visibility timeout.

16) Lower first-interaction latency to Lex

Question. Most relevant approach?

Intuition. First-hit slowness is usually **cold starts** on the API Lambda (LF0).

Reason. **Provisioned concurrency** keeps environments warm, reducing p99 latency for the initial call.

Answer. b) Provisioned concurrency on LF0.

17) Add Cognito Identity Pools for signed browser calls

Question. Which statement is correct?

Intuition. Identity Pools issue **temporary IAM creds** used by the AWS SDK to **SigV4** sign.

Reason. They don't require SES/verified emails; User Pools are auth, Identity Pools are for AWS creds.

Answer. a) Identity Pools issue temporary IAM creds for SigV4.

18) OpenSearch returns 403 though VPC/SGs are correct

Question. Next most probable cause?

Intuition. Network path works (no timeout). **403** implies **authorization** at OpenSearch, not networking.

Reason. Fix the domain **resource policy / FGAC** to allow the Lambda's **IAM role** (or its role mapping).

Answer. a) Missing OS resource policy/FGAC permission for the Lambda role.

19) Cut preflights without CloudFront (choose 2)

Question. Which two changes reduce preflights while keeping security?

Intuition. Preflights trigger on **non-simple** methods/headers/content types.

Reason. Use **simple headers** with Content-Type: application/x-www-form-urlencoded and restrict methods to **GET/POST**.

Answer. a) and d).

20) SQS → DLQ timing varies between tests (choose 2)

Question. Which pair explains variance?

Intuition. Two knobs govern DLQ timing: **visibility timeout** and **maxReceiveCount**; deletes vs. no-deletes also impact when counts increment.

Reason. Longer visibility delays retries; **maxReceiveCount** defines how many failed receives before DLQ.

Answer. a) and b).



Ask anything

+ Extended thinking ▾



ChatGPT can make mistakes. Check important info.