

AWS CLOUD DEVELOPER PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

**VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE**

<https://awsclouddev.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	9
Explanations	11
Next Steps	17

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

- 1. Which option describes a benefit that led the shipping industry and the software industry to adopt containers?**
 - A. The abstraction of details of what's being "shipped" increases the agility of a system.
 - B. Infrastructure providers can create strong dependencies between their platform and the types of containers that they choose to work with.
 - C. The ability to use nonstandard types of containers gives producers flexibility.
 - D. Increased visibility into the details of what's being "shipped" helps each point in the system to provide the right type of support.

- 2. Which constraint cannot be modified in AWS Lambda?**
 - A. Function timeout
 - B. Reserved concurrency
 - C. Burst quota
 - D. Regional quota

- 3. Which statement describes the relationship between resource APIs and service client APIs?**
 - A. Resource APIs provide a lower-level abstraction than service client APIs.
 - B. Resource APIs provide the same abstraction as service client APIs.
 - C. Resource APIs provide a higher-level abstraction than the low-level calls made by service client APIs.
 - D. Resource APIs do not exist in AWS.

- 4. Which statement about API Gateway integrations is true?**
 - A. A first-class integration connects an API Gateway HTTP API route to an external third-party service API route to an external third-party service API.
 - B. A developer can use HTTP proxy integrations to direct a route to a resource on the internet.
 - C. API Gateway cannot access resources within a virtual private cloud (VPC).
 - D. An API integration request encapsulates an HTTP request that the backend receives and is always different from the method request that the client submits.

- 5. How might a developer use AWS CodePipeline in their CI/CD process?**
- A. Host Git-based repositories and provide source control.
 - B. Deploy to a variety of compute services.
 - C. Compile source code, run tests, and produce software packages that are ready to deploy.
 - D. Automate the steps that are required to release software based on a release model that the developer defines.
- 6. What is the correct term for the process of verifying a user's identity?**
- A. Endorsement
 - B. Authorization
 - C. Confirmation
 - D. Authentication
- 7. Which practice about caching is NOT recommended?**
- A. Consider data access patterns and staleness tolerance
 - B. Add caching where data is slow to access
 - C. Add caching in as many places as possible early in development
 - D. Use multiple cache layers where appropriate
- 8. To reduce impact of failed records due to an older client version, what strategy might help?**
- A. Increase the visibility timeout and set a high value for MaxNumberOfMessages.
 - B. Add a dead-letter queue and set a low maxReceiveCount.
 - C. Use long polling and set a high wait time.
 - D. Change the queue type from Standard to FIFO and set a high throughput rate.
- 9. Which option is used to ensure atomic multi-item updates in DynamoDB?**
- A. DynamoDB transactions
 - B. DynamoDB Accelerator (DAX)
 - C. Global tables
 - D. Local secondary index

10. Which statement about Amazon Kinesis Data Streams is true?

- A. A standard data stream can have one consumer. With the enhanced fanout option, you can add multiple consumers.
- B. The data records in a data stream are distributed into shards.
- C. When a record is added to the stream, the record is available until a consumer deletes it.
- D. A stream consumer must delete the records that it processes successfully.

SAMPLE

Answers

SAMPLE

1. A
2. C
3. C
4. B
5. D
6. D
7. C
8. B
9. A
10. B

SAMPLE

Explanations

SAMPLE

1. Which option describes a benefit that led the shipping industry and the software industry to adopt containers?

- A. The abstraction of details of what's being "shipped" increases the agility of a system.**
- B. Infrastructure providers can create strong dependencies between their platform and the types of containers that they choose to work with.
- C. The ability to use nonstandard types of containers gives producers flexibility.
- D. Increased visibility into the details of what's being "shipped" helps each point in the system to provide the right type of support.

Abstraction and standardization are what made containers so appealing. By encapsulating an application and its runtime, libraries, and dependencies into a single, portable unit, containers decouple software from the specifics of the underlying infrastructure. Just like standard shipping containers let goods move smoothly between trucks, ships, and terminals, container images let software move between development laptops, on-prem servers, and clouds without reconfiguring for each environment. This consistency enables faster development, testing, deployment, and scaling, because you can trust that the same container runs the same way everywhere. The other ideas imply outcomes that would actually hinder portability or reliance on infrastructure specifics: tying the platform to particular container types creates lock-in; using nonstandard containers would break interoperability; and focusing on visibility into internal details runs counter to the portability and consistency containers aim to provide. That's why the abstraction of what's inside the container is the strongest driver of agility.

2. Which constraint cannot be modified in AWS Lambda?

- A. Function timeout
- B. Reserved concurrency
- C. Burst quota**
- D. Regional quota

Short-term scaling in Lambda relies on burst concurrency, which lets functions start many instances quickly during a spike. This burst capacity is an internal service mechanism and isn't something customers can directly configure or increase; it's determined by AWS based on usage patterns and region. You can change a function's timeout, you can set reserved concurrency for a function to guarantee a ceiling on its concurrent executions, and you can request increases to regional quotas. Because burst quota is managed by AWS and not adjustable by users, it's the constraint you can't modify. If you need predictable capacity, you can use provisioned concurrency to guarantee performance instead of relying on bursts.

3. Which statement describes the relationship between resource APIs and service client APIs?

- A. Resource APIs provide a lower-level abstraction than service client APIs.
- B. Resource APIs provide the same abstraction as service client APIs.
- C. Resource APIs provide a higher-level abstraction than the low-level calls made by service client APIs.**
- D. Resource APIs do not exist in AWS.

Resource APIs provide a higher-level, more convenient interface for working with AWS resources. They're built on top of the underlying service client calls and automate common patterns, turning multiple low-level operations into simple, resource-oriented actions. For example, in a typical SDK, you can interact with a bucket or table through object-like interfaces and methods (such as iterating over `bucket.objects` or performing a single put-like action), while the service client would require calling low-level operations (like `list_objects_v2`, `create_bucket`, etc.) and handling pagination, errors, and response parsing yourself. This is why resource APIs are described as offering a higher-level abstraction than the low-level calls made by service client APIs.

4. Which statement about API Gateway integrations is true?

- A. A first-class integration connects an API Gateway HTTP API route to an external third-party service API route to an external third-party service API.
- B. A developer can use HTTP proxy integrations to direct a route to a resource on the internet.**
- C. API Gateway cannot access resources within a virtual private cloud (VPC).
- D. An API integration request encapsulates an HTTP request that the backend receives and is always different from the method request that the client submits.

Understanding how API Gateway forwards requests to backends is key. An HTTP proxy integration routes the API route straight to an external HTTP endpoint, usually on the internet, using the same HTTP method and path unless you change them. This setup is ideal for quickly connecting your API Gateway to a third party service or any internet resource without extra backend coding, because the request is passed through to the target and the backend's response comes back through API Gateway to the client with minimal transformation. A term like "first-class integration" isn't a standard descriptor for API Gateway integrations, and API Gateway can access resources inside a VPC when you use private integrations or VPC Links, so the statement that it cannot access VPC resources isn't accurate. Also, the integration request doesn't have to be different from the client's method request; with a proxy integration it can pass through the same HTTP method and much of the original request, unless you choose to map or transform it.

5. How might a developer use AWS CodePipeline in their CI/CD process?

- A. Host Git-based repositories and provide source control.
- B. Deploy to a variety of compute services.
- C. Compile source code, run tests, and produce software packages that are ready to deploy.
- D. Automate the steps that are required to release software based on a release model that the developer defines.**

CodePipeline is about automating the release process itself. It lets you model a release workflow as a sequence of stages and actions and then run that workflow end to end, moving code from source to production in a repeatable, governed way. You define a release model with stages like source, build, test, and deploy, and attach actions such as compiling code, running tests, packaging artifacts, and deploying to environments. It can enforce gates and approvals, so releases only proceed when the defined criteria are met. This orchestration—and the ability to integrate with source control, build, and deployment tools (CodeCommit, GitHub, CodeBuild, CodeDeploy, ECS, Lambda, etc.)—is what makes CodePipeline the mechanism for delivering software automatically according to a developer-defined process.

6. What is the correct term for the process of verifying a user's identity?

- A. Endorsement
- B. Authorization
- C. Confirmation
- D. Authentication**

Authentication is the process of verifying a user's identity. When you log in, you prove who you are by presenting credentials you know (password), you possess (a code from an authenticator app or a hardware token), or you are (a biometric measure). The system uses these proofs to confirm the user's claimed identity before allowing access. Endorsement isn't about proving identity; it's more about giving approval or support for something. Authorization comes after authentication and determines what an authenticated user is allowed to do or access. Confirmation is a general term for verifying something, but in security contexts it isn't the specific term used for proving identity.

7. Which practice about caching is NOT recommended?

- A. Consider data access patterns and staleness tolerance
- B. Add caching where data is slow to access
- C. Add caching in as many places as possible early in development**
- D. Use multiple cache layers where appropriate

Caching works best when it is targeted to where it adds real value and respects how fresh the data must be. Adding caching in as many places as possible early in development leads to unnecessary complexity, since caches require invalidation logic, consistency handling, and tuning that may not be needed yet. Prematurely caching everywhere can mask underlying performance issues, waste effort, and make future changes harder. The best approach is to identify where data access is slow or expensive, understand how often data is read and how stale it can be, and then apply caching where it makes the most impact. If data retrieval is slow, caching can dramatically reduce latency. And using multiple cache layers—for example, an in-memory cache for ultra-fast access plus a distributed cache for larger, shared data—can provide a good balance of speed, capacity, and reliability.

8. To reduce impact of failed records due to an older client version, what strategy might help?

- A. Increase the visibility timeout and set a high value for `MaxNumberOfMessages`.
- B. Add a dead-letter queue and set a low `maxReceiveCount`.**
- C. Use long polling and set a high wait time.
- D. Change the queue type from Standard to FIFO and set a high throughput rate.

Using a dead-letter queue helps isolate and diagnose messages that repeatedly fail processing. In SQS, every failed attempt increments the receive count; when it reaches the configured `maxReceiveCount`, the message is moved to the dead-letter queue instead of being retried forever on the main queue. Setting a low `maxReceiveCount` means problematic records—such as those caused by an older client version—are redirected sooner, reducing their impact on normal processing and giving you a dedicated place to inspect, fix, or update the client before deciding whether to reprocess. The other options don't directly address failed processing: increasing the visibility timeout just delays retries, long polling affects wait time for messages, and changing queue type or throughput doesn't provide a mechanism to capture and handle failed records separately.

9. Which option is used to ensure atomic multi-item updates in DynamoDB?

- A. DynamoDB transactions**
- B. DynamoDB Accelerator (DAX)
- C. Global tables
- D. Local secondary index

Atomic multi-item updates in DynamoDB are achieved with DynamoDB transactions. Transactions let you group multiple Put, Update, and Delete operations across one or more tables into a single, all-or-nothing unit. If any operation fails or a conditional check fails, the entire group is rolled back, so no partial updates occur and consistency is preserved. You can include up to 25 actions in a single transaction, and you can also perform atomic reads in the same transaction context with a `TransactGetItems` call if needed. DynamoDB Accelerator (DAX) speeds reads by caching data in-memory, but it doesn't change how updates are applied or guarantee atomicity across multiple items. Global Tables enable multi-region replication, but they don't provide a single atomic transaction across regions. A Local Secondary Index is just an alternate sort-key index on the same table and has no role in coordinating multi-item updates. For a single item update, `UpdateItem` is atomic for that item, but to achieve atomicity across multiple items, a transaction is the right tool.

10. Which statement about Amazon Kinesis Data Streams is true?

- A. A standard data stream can have one consumer. With the enhanced fanout option, you can add multiple consumers.
- B. The data records in a data stream are distributed into shards.**
- C. When a record is added to the stream, the record is available until a consumer deletes it.
- D. A stream consumer must delete the records that it processes successfully.

In Amazon Kinesis Data Streams, data is organized into shards, which are the unit that provides throughput and parallelism. Each record is assigned to a shard (based on its partition key), and records within a shard are stored in order. The stream's overall capacity is the sum of the capacities of its shards, so you scale by adding or removing shards. Thus, the statement that data records in a data stream are distributed into shards is the best description of how Kinesis organizes data. It captures the fundamental idea that shards split the stream's data to enable parallel processing and predictable throughput. Why the other ideas aren't correct: a standard stream can have multiple consumers, though their read throughput may be shared unless you use enhanced fan-out, which gives each consumer its own dedicated throughput. Records aren't deleted when a consumer processes them; they remain in the stream for a retention period (configurable, not deleted by the consumer). There's no mechanism where a consumer must delete records after processing.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://awsclouddev.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE