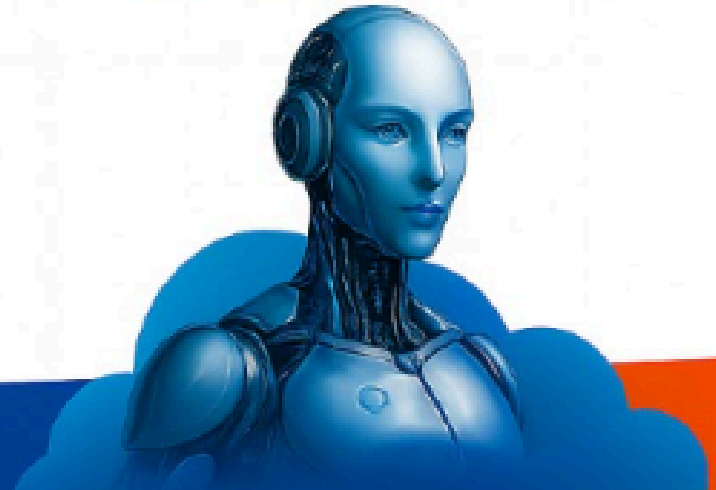
aws

AWS CERTIFIED MACHINE LEARNING ENGINEER

ASSOCIATE



MLA-C01

PRACTICE EXAM QUESTIONS

CloudCertification Store

AWS Certified Machine Learning Engineer Associate MLA-C01 - Practice Exam Questions (AWS-MLA-C01-0010)

DISCLAIMER

- This practice test includes questions **compiled from various exam preparation platforms.**
- Important: **Questions and answers were AI-assisted and human-curated**. Verify accuracy with official documentation before relying on this material.
- **Users are strongly encouraged to** double-check all content against **official documentation and trusted sources** before using it for exam preparation or making important decisions.
- The creators of this material assume **no responsibility** for any errors, inaccuracies, or outcomes, including exam results, based on the use of this content.
- **Some questions might be duplicated or close** to previous ones, this is done on purpose as a way to re-inforce your learning.
- Single-user licence only
    - Includes one unique Payhip Licence Key per purchase, along with a Product Key.
    - Redistribution, resale, or public posting is prohibited. We can trace any file to the purchaser, with the use of the purchased License Key and Product Key.

AWS Certified Machine Learning Engineer Associate MLA-C01 - Practice Exam Questions (AWS-MLA-C01-0010)

AWS Certified Machine Learning Engineer – Associate

Issued by Amazon Web Services Training and Certification

Earners of this badge have knowledge and skills in developing, deploying, maintaining, and monitoring ML solutions to meet AI/ML objectives. They know how to ingest, transform, validate, and prepare data for ML modeling. They have skills in implementing and operationalizing ML workloads in production. They can select modeling approaches and analyze model performance. They have the expertise to monitor ML solutions and to secure ML systems and resources.

https://aws.amazon.com/certification/certified-machine-learning-engineer-associate/

Exam overview

AWS Certified Machine Learning Engineer - Associate

Category

Associate

Exam duration

130 minutes

Exam format

65 questions

Cost

150 USD. Visit Exam pricing for additional cost information, including foreign exchange rates

Intended candidate

Individuals with at least 1 year of experience using Amazon SageMaker and other ML engineering AWS services

Candidate role examples

Backend software developer, DevOps engineer, data engineer, MLOps engineer, and data scientist

Testing options

Pearson VUE testing center or online proctored exam

Prepare for the exam

Go from start to certified. Follow our Exam Prep Plan on AWS Skill Builder, our online learning center, so you can approach exam day with confidence.

1. Get to know the exam with exam-style questions

Follow the 4-step plan.

Review the exam guide.

Take the AWS Certification Official Practice Question Set to understand exam-style questions.

Take the AWS Certification Official Pretest to identify any areas where you need to refresh your AWS knowledge and skills.

2. Refresh your AWS knowledge and skills

Enroll in digital courses where you need to fill gaps in knowledge and skills, practice with AWS Builder Labs, AWS Cloud Quest, and AWS Jam.

3. Review and practice for your exam

Review the scope of the exam. Explore each exam domain's topics and how they align to AWS services. Reinforce your knowledge and identify learning gaps with exam-style questions and flashcards. Follow instructors as they walk through exam-style questions and provide test-taking strategies. Continue practicing with AWS Builder Labs and/or AWS SimuLearn.

4. Assess your exam readiness

Take the AWS Certification Official Practice Exam.

Key FAQs to help you get started

Who should earn AWS Certified Machine Learning Engineer - Associate?

The ideal candidate for this exam has at least 1 year of experience in machine learning engineering or a related field and 1 year of hands-on experience with AWS services. Professionals who do not have prior machine learning experience can take the training available in the Exam Prep Plans and get started building their knowledge and skills.

How will the AWS Certified Machine Learning Engineer - Associate help my career?

Per the World Economic Forum Future of Jobs Report 2023, demand for AI and Machine Learning Specialists is expected to grow by 40%. However, 70% of North American IT leaders say they have the greatest difficulty filling AI/ML specialist roles. This certification can position you for in-demand machine learning jobs in AWS Cloud.

How is AWS Certified Machine Learning Engineer - Associate different from AWS Certified Machine Learning - Specialty?

AWS Certified Machine Learning Engineer - Associate is a role-based certification designed for ML engineers and MLOps engineers with at least one year of experience in AI/ML.

AWS Machine Learning - Specialty is a specialty certification covering topics across data engineering, data analysis, modeling, and ML implementation and ops. It is more suitable for individuals with 2 or more years of experience developing, architecting, and running ML workloads on AWS.

What certification(s) should I earn next after AWS Certified Machine Learning Engineer - Associate?

For professionals looking to dive deeper into machine learning, we recommend AWS Certified Machine Learning - Specialty.

How long is this certification valid for?

This certification is valid for 3 years. Before your certification expires, you can recertify by passing the latest version of this exam. Learn more about recertification options for AWS Certifications.

Additional resources

Learn more about AWS Certification exams

Before scheduling your AWS Certification exam, review the available options for the specific exam and your desired exam language.

View all exams

AWS Training Live on Twitch

Access free, live, and on-demand training on our dedicated Twitch channel. Join AWS experts for live shows, chat with the community, or explore on-demand training.

Explore more

AWS Certification FAQs

Questions about AWS Certification? Browse frequently asked questions about getting AWS Certified and AWS Certification.

Browse AWS Certification FAQ

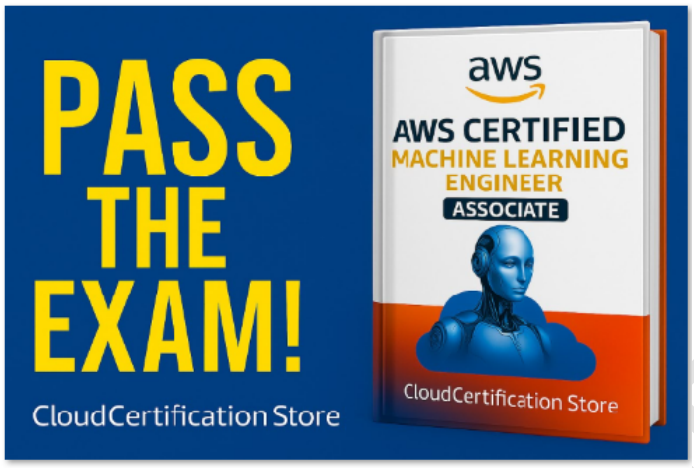
Information and policies

Learn about what to expect with scheduling your exam, identification requirements, exam functionality, relevant policies, and more.
Explore more

Exam vouchers

Are you supporting a team taking AWS Certification exams? Make it easier with exam vouchers. Purchase online anytime, and then use a self-service portal to efficiently distribute, track, and manage standard exam vouchers.

Get exam vouchers



Practice Questions

Question 1

An ML engineer needs to use data with Amazon SageMaker Canvas to train an ML model. The data is stored in Amazon S3 and is complex in structure. The ML engineer must use a file format that minimizes processing time for the data.

Which file format will meet these requirements?

A. CSV files compressed with Snappy

B. JSON objects in JSONL format

C. JSON files compressed with gzip

D. Apache Parquet files

✅ **Correct answer: D. Apache Parquet files**

Parquet is a columnar, compressed, and splittable format that minimizes I/O and speeds up reads for analytical workloads, which is ideal for complex, nested structures. Canvas and downstream AWS analytics services optimize for Parquet, reducing the time spent parsing and transforming records. Column pruning and predicate pushdown further reduce processing time when only a subset of

columns is needed. This combination yields faster data ingest and transformation than row-oriented formats like CSV or JSON.

Incorrect answers:

❌ **A. CSV files compressed with Snappy** – CSV is a row-oriented, text-based format that lacks schema information and efficient columnar access, so it incurs high parsing overhead. While Snappy helps with compression/decompression speed, it does not fix CSV's lack of type fidelity nor enable predicate pushdown. CSV also struggles with complex or nested data, requiring additional parsing logic. Overall, it results in more CPU time and slower ingestion for complex datasets.

❌ **B. JSON objects in JSONL format** – JSONL preserves one JSON object per line, which is convenient for streaming but still row-oriented and verbose. Complex nested structures are supported, but the lack of columnar layout means no column pruning and heavier parsing costs. Compression is not inherent and, if applied, increases CPU time for decompression. As a result, JSONL typically increases processing time compared to Parquet.

❌ **C. JSON files compressed with gzip** – Gzip compression reduces storage, but decompression is CPU-intensive and not splittable by default, hindering parallelism on large files. JSON is row-oriented and verbose, further increasing parse time for complex structures. There is no native schema enforcement, which

can add overhead in downstream processing. Together these factors make gzip-compressed JSON slower than Parquet for Canvas workflows.

Question 2

A company is using an AWS Lambda function to monitor the metrics from an ML model. An ML engineer needs to implement a solution to send an email message when the metrics breach a threshold.

Which solution will meet this requirement?

A. Log the metrics from the Lambda function to AWS CloudTrail. Configure a CloudTrail trail to send the email message.

B. Log the metrics from the Lambda function to Amazon CloudFront. Configure an Amazon CloudWatch alarm to send the email message.

C. Log the metrics from the Lambda function to Amazon CloudWatch. Configure a CloudWatch alarm to send the email message.

D. Log the metrics from the Lambda function to Amazon CloudWatch. Configure an Amazon CloudFront rule to send the email message.

✅ Correct answer: C. Log the metrics from the Lambda function to Amazon CloudWatch. Configure a CloudWatch alarm to send the email message.

CloudWatch is the native destination for Lambda metrics and custom application metrics, enabling immediate alarm creation. CloudWatch Alarms integrate with Amazon SNS to send email notifications when thresholds are breached. This approach is fully managed and requires minimal code: emit metrics and define an alarm with an SNS subscription. It also provides dashboards and history for easy operational visibility.

Incorrect answers:

❌ **A. Log the metrics from the Lambda function to AWS CloudTrail. Configure a CloudTrail trail to send the email message.** – CloudTrail records control-plane API calls, not arbitrary application metrics. It is not designed for metric thresholds or real-time metric evaluation. You cannot create metric alarms directly from CloudTrail events for this use case. This would be a misuse of the service and add unnecessary complexity.

❌ **B. Log the metrics from the Lambda function to Amazon CloudFront. Configure an Amazon CloudWatch alarm to send the email message.** – CloudFront is a content delivery network and does not serve as a general-purpose metrics sink. While CloudFront has its own metrics, it's unrelated to Lambda application metrics. Routing metrics through CloudFront adds no value and

introduces confusion. CloudWatch remains the correct place for metrics and alarms.

❌ **D. Log the metrics from the Lambda function to Amazon CloudWatch. Configure an Amazon CloudFront rule to send the email message.** – Even with metrics in CloudWatch, CloudFront has no role in sending notifications for metric breaches. CloudWatch Alarms with SNS subscriptions are the supported mechanism for email alerts. Introducing CloudFront offers no benefit and complicates the design. The simplest, standard path is CloudWatch → Alarm → SNS (email).

Question 3

An ML engineer needs to implement a solution to host a trained ML model. The rate of requests to the model will be inconsistent throughout the day.

The ML engineer needs a scalable solution that minimizes costs when the model is not in use. The solution also must maintain the model's capacity to respond to requests during times of peak usage.

Which solution will meet these requirements?

A. Create AWS Lambda functions that have fixed concurrency to host the model. Configure the Lambda functions to automatically scale based on the number of requests to the model.

B. Deploy the model on an Amazon Elastic Container Service (Amazon ECS) cluster that uses AWS Fargate. Set a static number of tasks to handle requests during times of peak usage.

C. Deploy the model to an Amazon SageMaker endpoint. Deploy multiple copies of the model to the endpoint. Create an Application Load Balancer to route traffic between the different copies of the model at the endpoint.

D. Deploy the model to an Amazon SageMaker endpoint. Create SageMaker endpoint auto scaling policies that are based on Amazon CloudWatch metrics to adjust the number of instances dynamically.

✅ **Correct answer: D. Deploy the model to an Amazon SageMaker endpoint. Create SageMaker endpoint auto scaling policies that are based on Amazon CloudWatch metrics to adjust the number of instances dynamically.**

SageMaker real-time endpoints support automatic scaling based on utilization metrics (e.g., invocations per instance), preserving performance under load and reducing capacity when idle. This directly addresses fluctuating traffic while minimizing cost. It also avoids operational overhead by letting the service manage

scaling rather than maintaining custom logic. The approach remains compatible with production features like Model Monitor and multi-variant testing.

Incorrect answers:

❌ **A. Create AWS Lambda functions that have fixed concurrency to host the model. Configure the Lambda functions to automatically scale based on the number of requests to the model.** – Lambda is not ideal for stateful model servers or large containerized ML runtimes, and "fixed concurrency" contradicts the need to scale elastically. Packaging heavy frameworks into Lambda often leads to cold start challenges and deployment complexity. You would still need to build custom monitoring and autoscaling behavior. SageMaker provides a purpose-built, managed inference platform.

❌ **B. Deploy the model on an Amazon Elastic Container Service (Amazon ECS) cluster that uses AWS Fargate. Set a static number of tasks to handle requests during times of peak usage.** – A static task count wastes resources during off-peak periods and fails to minimize costs when traffic is low. While ECS with Fargate can scale, the option explicitly fixes capacity, which violates the requirement. You would also need to build additional observability and autoscaling policies manually. SageMaker's integrated scaling is simpler and more targeted.

❌ **C. Deploy the model to an Amazon SageMaker endpoint. Deploy multiple copies of the model to the endpoint. Create an Application Load Balancer to route traffic between the different copies of the model at the endpoint.** – SageMaker endpoints already handle load balancing and scaling, so adding an ALB is redundant and adds complexity. Manually managing "copies" does not guarantee elasticity or cost efficiency. Autoscaling in SageMaker is the supported pattern for demand variability. This answer increases operational overhead without benefits.

Question 4

A company has a binary classification model in production. An ML engineer needs to develop a new version of the model.

The new model version must maximize correct predictions of positive labels and negative labels. The ML engineer must use a metric to recalibrate the model to meet these requirements.

Which metric should the ML engineer use for the model recalibration?

A. Accuracy

B. Precision

C. Recall

D. Specificity

✅ **Correct answer: A. Accuracy**

Accuracy directly measures the proportion of all correct predictions, combining both true positives and true negatives. If the goal is to maximize correctness across both classes, optimizing thresholding by accuracy aligns with that objective. While not always ideal for imbalanced datasets, the prompt explicitly emphasizes correct positives and negatives together. Thus, accuracy serves as the most straightforward recalibration metric here.

Incorrect answers:

❌ **B. Precision** – Precision focuses on the quality of positive predictions (TP / (TP + FP)) and ignores how well negatives are identified. Optimizing solely for precision could reduce false positives at the expense of missing many true positives. This does not align with maximizing correctness across both classes. It provides a partial view rather than a holistic measure.

❌ **C. Recall** – Recall (TP / (TP + FN)) emphasizes capturing as many positives as possible, potentially increasing false positives. Maximizing recall may degrade performance on negative classifications, hurting overall correctness. If both

positive and negative correctness matter equally, recall alone is insufficient. It's a targeted metric, not a balanced one.

❌ **D. Specificity** – Specificity (TN / (TN + FP)) focuses only on correctly identifying negatives. While valuable in certain domains, it ignores positive classification performance. Maximizing specificity might severely lower recall, which undermines the "maximize correct positives and negatives" requirement. It is not an all-encompassing correctness metric like accuracy.

Question 5

An ML engineer needs to deploy ML models to get inferences from large datasets in an asynchronous manner. The ML engineer also needs to implement scheduled monitoring of the data quality of the models. The ML engineer must receive alerts when changes in data quality occur.

Which solution will meet these requirements?

A. Deploy the models by using scheduled AWS Glue jobs. Use Amazon CloudWatch alarms to monitor the data quality and to send alerts.

B. Deploy the models by using scheduled AWS Batch jobs. Use AWS CloudTrail to monitor the data quality and to send alerts.

C. Deploy the models by using Amazon Elastic Container Service (Amazon ECS) on AWS Fargate. Use Amazon EventBridge to monitor the data quality and to send alerts.

D. Deploy the models by using Amazon SageMaker batch transform. Use SageMaker Model Monitor to monitor the data quality and to send alerts.

✅ **Correct answer: D. Deploy the models by using Amazon SageMaker batch transform. Use SageMaker Model Monitor to monitor the data quality and to send alerts.**

Batch transform is designed for asynchronous, large-scale inference without managing endpoint uptime. SageMaker Model Monitor natively profiles data quality on a schedule and integrates with CloudWatch/SNS for alerts. This pair provides an end-to-end, managed solution with minimal operational overhead. It avoids building custom pipelines for both inference scheduling and data drift detection.

Incorrect answers:

❌ **A. Deploy the models by using scheduled AWS Glue jobs. Use Amazon CloudWatch alarms to monitor the data quality and to send alerts.** – Glue is an ETL service, not an ML inference service, so you would be crafting nonstandard inference logic within ETL jobs. CloudWatch alarms require you to define and

emit your own quality metrics, increasing custom work. There is no built-in ML data quality profiling. This adds complexity compared to Model Monitor's native capabilities.

❌ **B. Deploy the models by using scheduled AWS Batch jobs. Use AWS CloudTrail to monitor the data quality and to send alerts.** – While Batch can orchestrate containers, it lacks ML-specific features for data quality. CloudTrail tracks API calls, not dataset characteristics or statistical drift, so it's unsuitable for data quality monitoring. You would need to implement comprehensive custom checks and alerting. This is higher effort and less maintainable.

❌ **C. Deploy the models by using Amazon Elastic Container Service (Amazon ECS) on AWS Fargate. Use Amazon EventBridge to monitor the data quality and to send alerts.** – ECS can run inference containers, but EventBridge is an event router, not a data quality analyzer. You would still need to compute drift statistics, thresholds, and alerting logic. This shifts significant engineering burden onto your team. SageMaker's managed services are purpose-built for these needs.

Question 6

A company has an ML model that needs to run one time each night to predict stock values. The model input is 3 MB of data that is collected during the current day. The model produces the predictions for the next day. The prediction process takes less than 1 minute to finish running.

How should the company deploy the model on Amazon SageMaker to meet these requirements?

A. Use a multi-model serverless endpoint. Enable caching.

B. Use an asynchronous inference endpoint. Set the InitialInstanceCount parameter to 0.

C. Use a real-time endpoint. Configure an auto scaling policy to scale the model to 0 when the model is not in use.

D. Use a serverless inference endpoint. Set the MaxConcurrency parameter to 1.

✅ **Correct answer: D. Use a serverless inference endpoint. Set the MaxConcurrency parameter to 1.**

Serverless inference removes idle capacity cost when the endpoint is not invoked, which suits a once-per-night job. With a very short runtime and tiny input, a minimal concurrency setting is sufficient and cost-efficient. You avoid managing instances or scaling policies entirely. Cold starts are negligible here given the single, scheduled invocation window.

Incorrect answers:

❌ **A. Use a multi-model serverless endpoint. Enable caching.** – Multi-model endpoints are beneficial when hosting many models behind one endpoint, which is not required here. Caching does not materially improve a once-per-day, sub-minute job with small input. This adds unnecessary complexity without cost or performance benefits. A single serverless endpoint is simpler and cheaper.

❌ **B. Use an asynchronous inference endpoint. Set the InitialInstanceCount parameter to 0.** – Asynchronous endpoints are designed for larger payloads or long-running requests with queueing, which is overkill for a 1-minute nightly job. Also, asynchronous endpoints do not use "InitialInstanceCount" in the way real-time variants do. Serverless provides the simplest pay-per-invocation model. This option adds operational features you don't need.

❌ **C. Use a real-time endpoint. Configure an auto scaling policy to scale the model to 0 when the model is not in use.** – Real-time endpoints do not natively scale down to zero; you would pay for idle capacity. Managing lifecycle (start/stop) to emulate zero is additional overhead. For a nightly run, continuous provisioning is wasteful. Serverless avoids these costs and management tasks.

Question 7

An advertising company uses AWS Lake Formation to manage a data lake. The data lake contains structured data and unstructured data. The company's ML engineers are assigned to specific advertisement campaigns.

The ML engineers must interact with the data through Amazon Athena and by browsing the data directly in an Amazon S3 bucket. The ML engineers must have access to only the resources that are specific to their assigned advertisement campaigns.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Configure IAM policies on an AWS Glue Data Catalog to restrict access to Athena based on the ML engineers' campaigns.

B. Store users and campaign information in an Amazon DynamoDB table. Configure DynamoDB Streams to invoke an AWS Lambda function to update S3 bucket policies.

C. Use Lake Formation to authorize AWS Glue to access the S3 bucket. Configure Lake Formation tags to map ML engineers to their campaigns.

D. Configure S3 bucket policies to restrict access to the S3 bucket based on the ML engineers' campaigns.

✅ **Correct answer: C. Use Lake Formation to authorize AWS Glue to access the S3 bucket. Configure Lake Formation tags to map ML engineers to their campaigns.**

Lake Formation provides fine-grained, tag-based access control across both cataloged tables (for Athena) and underlying S3 data locations. By using LF-tags tied to campaigns, you centralize and scale permissions management without handcrafting policies per user or dataset. This keeps governance consistent for both query and direct S3 access. It's the native, operationally efficient path for campaign-scoped permissions in a data lake.

Incorrect answers:

❌ **A. Configure IAM policies on an AWS Glue Data Catalog to restrict access to Athena based on the ML engineers' campaigns.** – IAM alone does not easily express fine-grained, column/table-level entitlements across evolving datasets. It also doesn't propagate smoothly to S3 object-level controls for the same logical datasets. Maintaining bespoke IAM for every campaign increases complexity. Lake Formation's tag-based model is purpose-built for this.

❌ **B. Store users and campaign information in an Amazon DynamoDB table. Configure DynamoDB Streams to invoke an AWS Lambda function to update S3 bucket policies.** – This creates a custom entitlement system you must build,

test, and maintain. It doesn't natively integrate with Athena's view of data, risking policy drift between S3 and the catalog. Automated policy updates add operational fragility. Lake Formation already solves this use case without custom code.

❌ **D. Configure S3 bucket policies to restrict access to the S3 bucket based on the ML engineers' campaigns.** – S3 bucket policies alone are coarse and hard to maintain for campaign-level granularity across many prefixes/objects. You also need to align permissions with Athena's Data Catalog to avoid mismatches. This approach quickly becomes unmanageable at scale. Lake Formation unifies governance across query and storage layers.

Question 8

A company's ML engineer has deployed an ML model for sentiment analysis to an Amazon SageMaker endpoint. The ML engineer needs to explain to company stakeholders how the model makes predictions.

Which solution will provide an explanation for the model's predictions?

A. Use SageMaker Model Monitor on the deployed model.

B. Use SageMaker Clarify on the deployed model.

C. Show the distribution of inferences from A/B testing in Amazon CloudWatch.

D. Add a shadow endpoint. Analyze prediction differences on samples.

✅ **Correct answer: B. Use SageMaker Clarify on the deployed model.**

SageMaker Clarify provides model explainability reports, including SHAP-based feature attributions that show how inputs influence predictions. It integrates directly with endpoints to generate explanations on production models. These insights are designed for stakeholders to understand and trust model behavior. Clarify also supports bias analysis, enhancing governance and transparency.

Incorrect answers:

❌ **A. Use SageMaker Model Monitor on the deployed model.** – Model Monitor focuses on data quality and drift, not per-prediction explainability. It won't tell stakeholders which features drove a specific outcome. While useful for operations, it doesn't answer "why" a particular prediction occurred. Clarify is the correct tool for explanations.

❌ **C. Show the distribution of inferences from A/B testing in Amazon CloudWatch.** – A/B distributions provide performance comparisons, not feature attributions. Stakeholders still won't know how inputs affected a single

prediction. This approach lacks transparency into model internals. It's complementary for experiments, not for explainability.

❌ **D. Add a shadow endpoint. Analyze prediction differences on samples.** – Shadow testing compares models under real traffic but does not expose how a model reasons. You would only see output divergences, not the feature contributions. It's an evaluation tactic, not an explainability method. Clarify directly addresses the requirement.

Question 9

A company wants to reduce the cost of its containerized ML applications. The applications use ML models that run on Amazon EC2 instances, AWS Lambda functions, and an Amazon Elastic Container Service (Amazon ECS) cluster. The EC2 workloads and ECS workloads use Amazon Elastic Block Store (Amazon EBS) volumes to save predictions and artifacts.

An ML engineer must identify resources that are being used inefficiently. The ML engineer also must generate recommendations to reduce the cost of these resources.

Which solution will meet these requirements with the LEAST development effort?

A. Create code to evaluate each instance's memory and compute usage.

B. Add cost allocation tags to the resources. Activate the tags in AWS Billing and Cost Management.

C. Check AWS CloudTrail event history for the creation of the resources.

D. Run AWS Compute Optimizer.

✅ **Correct answer: D. Run AWS Compute Optimizer.**

Compute Optimizer analyzes utilization metrics across EC2, Lambda, EBS volumes, and ECS on Fargate to surface right-sizing and configuration recommendations. It provides actionable guidance (e.g., instance families, volume types/sizes) without writing custom analysis code. This directly targets cost inefficiencies with minimal setup. The recommendations help you realize savings quickly across multiple compute modalities.

Incorrect answers:

❌ **A. Create code to evaluate each instance's memory and compute usage.** – Building a custom telemetry and analysis pipeline is time-consuming and error-prone. You would need to ingest CloudWatch metrics, implement heuristics, and maintain dashboards. This duplicates capabilities that Compute Optimizer already provides. It raises operational overhead for limited benefit.

❌ **B. Add cost allocation tags to the resources. Activate the tags in AWS Billing and Cost Management.** – Tags improve cost visibility and chargeback but do not generate optimization recommendations. You would still need to analyze usage manually to find inefficiencies. While tagging is a good practice, it does not satisfy the requirement to identify and recommend right-sizing actions. Compute Optimizer addresses that gap directly.

❌ **C. Check AWS CloudTrail event history for the creation of the resources.** – CloudTrail reveals provision events, not ongoing utilization or cost efficiency. Creation history doesn't indicate whether a resource is oversized or underutilized. This provides little to no guidance for optimization. It's not the right tool for cost reduction analysis.

Question 10

A company has deployed an ML model that detects fraudulent credit card transactions in real time in a banking application. The model uses Amazon SageMaker Asynchronous Inference. Consumers are reporting delays in receiving the inference results.

An ML engineer needs to implement a solution to improve the inference performance. The solution also must provide a notification when a deviation in model quality occurs.

Which solution will meet these requirements?

A. Use SageMaker real-time inference for inference. Use SageMaker Model Monitor for notifications about model quality.

B. Use SageMaker batch transform for inference. Use SageMaker Model Monitor for notifications about model quality.

C. Use SageMaker Serverless Inference for inference. Use SageMaker Inference Recommender for notifications about model quality.

D. Keep using SageMaker Asynchronous Inference for inference. Use SageMaker Inference Recommender for notifications about model quality.

✅ **Correct answer: A. Use SageMaker real-time inference for inference. Use SageMaker Model Monitor for notifications about model quality.**

Real-time endpoints deliver low-latency predictions suitable for interactive fraud detection and eliminate queue-induced lag from asynchronous processing. Model Monitor can continuously evaluate production data quality and emit alerts when drift or schema violations occur. This pairing directly addresses both latency and

quality monitoring requirements. It also avoids building custom alerting or data checks.

Incorrect answers:

❌ **B. Use SageMaker batch transform for inference. Use SageMaker Model Monitor for notifications about model quality.** – Batch transform is designed for offline, large-scale jobs and cannot meet interactive latency demands. It would increase, not decrease, response time for consumers. While Model Monitor is still valid, the inference mode is inappropriate. Real-time inference is the proper fit for immediate decisions.

❌ **C. Use SageMaker Serverless Inference for inference. Use SageMaker Inference Recommender for notifications about model quality.** – Serverless inference may help occasionally, but it can introduce cold starts and is not guaranteed to resolve latency concerns under bursty real-time loads. Inference Recommender evaluates configuration/performance, not model quality drift or data issues. This combination does not ensure low latency and the right alerting. Real-time endpoints plus Model Monitor is the targeted solution.

❌ **D. Keep using SageMaker Asynchronous Inference for inference. Use SageMaker Inference Recommender for notifications about model quality.** – Asynchronous inference is the cause of delays due to queueing semantics and

background processing. Inference Recommender won't notify on data quality deviations and is not intended for production monitoring. This option fails to address both the latency and the monitoring needs. Switching to real-time and enabling Model Monitor resolves the core issues.

Question 11

An ML engineer needs to use AWS CloudFormation to create an ML model that an Amazon SageMaker endpoint will host.

Which resource should the ML engineer declare in the CloudFormation template to meet this requirement?

A. AWS::SageMaker::Model

B. AWS::SageMaker::Endpoint

C. AWS::SageMaker::NotebookInstance

D. AWS::SageMaker::Pipeline

✅ **Correct answer: A. AWS::SageMaker::Model**

This resource defines the model artifacts and the container image configuration that SageMaker endpoints consume. In SageMaker, you first create a Model resource, then attach it to a hosted endpoint configuration and endpoint for

inference. Declaring the model in CloudFormation enables repeatable, version-controlled deployments. Without the Model resource, the endpoint would have nothing to serve.

Incorrect answers:

❌ **B. AWS::SageMaker::Endpoint** – An endpoint is the hosting resource but it references a Model (via an EndpointConfig). Creating only an endpoint without a defined Model does not specify the artifacts or container. Endpoints are typically created after a Model and EndpointConfig exist. Alone, it cannot satisfy the "create an ML model" requirement.

❌ **C. AWS::SageMaker::NotebookInstance** – Notebooks are for interactive development, not for defining hosted model artifacts. Spinning up a notebook does not deploy a model to an endpoint or register model containers. It provides compute for exploration rather than production inference resources.

❌ **D. AWS::SageMaker::Pipeline** – A Pipeline orchestrates ML steps (processing, training, model registration), but it does not directly define the runtime model object an endpoint needs. You still end up creating a Model resource as a step output. Pipelines alone cannot host inference without a Model declaration.

Question 12

An ML engineer is evaluating several ML models and must choose one model to use in production. The cost of false negative predictions by the models is much higher than the cost of false positive predictions.

Which metric finding should the ML engineer prioritize the MOST when choosing the model?

A. Low precision

B. High precision

C. Low recall

D. High recall

✅ **Correct answer: D. High recall**

High recall means the model captures the majority of actual positives, minimizing false negatives. When missing a positive case is much costlier than raising extra alerts, recall is the appropriate optimization target. Prioritizing recall reduces the risk of overlooking true positive events. Thresholds can later be tuned to balance precision within acceptable business limits.

Incorrect answers:

❌ **A. Low precision** – Low precision implies many false positives, which may be acceptable only if recall is very high, but "low precision" is not a desirable property. It increases unnecessary follow-up actions and noise for downstream teams. It's not a metric you would intentionally prioritize as "low." The problem statement emphasizes reducing false negatives, not encouraging false positives outright.

❌ **B. High precision** – High precision minimizes false positives, which is secondary in this scenario. Optimizing precision typically comes at the cost of recall, potentially missing true positives. That trade-off contradicts the requirement to avoid false negatives. Precision can be addressed after recall targets are met.

❌ **C. Low recall** – Low recall explicitly means many false negatives, which is the opposite of the goal. A model with low recall would routinely miss positive cases. This would create unacceptable business risk. Such a model should be deprioritized for this use case.

Question 13

A company is using an Amazon Redshift database as its single data source. Some of the data is sensitive.

A data scientist needs to use some of the sensitive data from the database. An ML engineer must give the data scientist access to the data without transforming the source data and without storing anonymized data in the database.

Which solution will meet these requirements with the LEAST implementation effort?

A. Configure dynamic data masking policies to control how sensitive data is shared with the data scientist at query time.

B. Create a materialized view with masking logic on top of the database. Grant the necessary read permissions to the data scientist.

C. Unload the Amazon Redshift data to Amazon S3. Use Amazon Athena to create schema-on-read with masking logic. Share the view with the data scientist.

D. Unload the Amazon Redshift data to Amazon S3. Create an AWS Glue job to anonymize the data. Share the dataset with the data scientist.

✅ **Correct answer: A. Configure dynamic data masking policies to control how sensitive data is shared with the data scientist at query time.**

Dynamic data masking allows policy-based redaction at query time, avoiding data movement and additional storage. It’s purpose-built to share sensitive data

responsibly with minimal engineering overhead. The source remains authoritative, and policies can be adjusted centrally without duplicating data. This meets both “no transform” and “no stored anonymized copy” constraints efficiently.

Incorrect answers:

❌ **B. Create a materialized view with masking logic** – Materialized views create and store derived data, which adds management overhead and may not align with “no storing anonymized data.” They also introduce refresh considerations and latency between source and view. This is heavier operationally than dynamic masking. It’s not the least-effort approach.

❌ **C. Unload to S3 and use Athena** – Exporting data and building schema-on-read masking in Athena breaks single-source authority and creates data sprawl. It also adds pipelines, permissions, and lifecycle considerations. This contradicts the desire to avoid transforming or storing anonymized data elsewhere. It’s more complex and riskier than in-database masking.

❌ **D. Unload to S3 and anonymize with Glue** – This explicitly stores an anonymized copy, violating the requirement. It introduces ETL jobs, code, and operational maintenance. Data drift between copies becomes a concern. It’s the opposite of a “least effort” control.

Question 14

A company has AWS Glue data processing jobs that are orchestrated by an AWS Glue workflow. The AWS Glue jobs can run on a schedule or can be launched manually.

The company is developing pipelines in Amazon SageMaker Pipelines for ML model development. The pipelines will use the output of the AWS Glue jobs during the data processing phase of model development. An ML engineer needs to implement a solution that integrates the AWS Glue jobs with the pipelines.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use AWS Step Functions for orchestration of the pipelines and the AWS Glue jobs.

B. Use processing steps in SageMaker Pipelines. Configure inputs that point to the Amazon R esource Names (ARNs) of the AWS Glue jobs.

C. Use Callback steps in SageMaker Pipelines to start the AWS Glue workflow and to stop the pipelines until the AWS Glue jobs finish running.

D. Use Amazon EventBridge to invoke the pipelines and the AWS Glue jobs in the desired order.

✅ **Correct answer: C. Use Callback steps in SageMaker Pipelines to start the AWS Glue workflow and to stop the pipelines until the AWS Glue jobs finish running.**

Callback steps are designed to integrate external systems into a pipeline while preserving pipeline state and dependencies. They let you trigger a Glue workflow and block until an external completion signal arrives, keeping orchestration inside SageMaker. This yields tighter control, simpler monitoring, and less moving parts. It avoids building and maintaining a separate orchestrator.

Incorrect answers:

❌ **A. Use AWS Step Functions** – While Step Functions can orchestrate both services, it introduces another control plane and duplicate orchestration logic. You would manage state transitions across two systems, which increases complexity. The question asks for least operational overhead. Callback steps keep it native to Pipelines.

❌ **B. Use processing steps with Glue job ARNs** – Processing steps execute containers managed by SageMaker, not Glue jobs, and cannot directly "point at" Glue jobs to run them. This misunderstands the purpose of a processing step. You would still need a mechanism to trigger and await Glue completion. It's not a valid integration.

❌ **D. Use Amazon EventBridge to invoke both** – EventBridge can fan-out triggers, but coordinating run order and completion semantics becomes harder. You'd have to build custom patterns and idempotency checks. It's looser coupling and higher effort than a single pipeline with a callback step.

Question 15

A company stores time-series data about user clicks in an Amazon S3 bucket. The raw data consists of millions of rows of user activity every day. ML engineers access the data to develop their ML models.

The ML engineers need to generate daily reports and analyze click trends over the past 3 days by using Amazon Athena. The company must retain the data for 30 days before archiving the data.

Which solution will provide the HIGHEST performance for data retrieval?

A. Keep all the time-series data without partitioning in the S3 bucket. Manually move data that is older than 30 days to separate S3 buckets.

B. Create AWS Lambda functions to copy the time-series data into separate S3 buckets. Apply S3 Lifecycle policies to archive data that is older than 30 days to S3 Glacier Flexible R etrieval.

C. Organize the time-series data into partitions by date prefix in the S3 bucket.

Apply S3 Lifecycle policies to archive partitions that are older than 30 days to S3 Glacier Flexible R etrieval.

D. Put each day's time-series data into its own S3 bucket. Use S3 Lifecycle policies to archive S3 buckets that hold data that is older than 30 days to S3 Glacier Flexible R etrieval.

✅ **Correct answer: C. Organize the time-series data into partitions by date prefix in the S3 bucket. Apply S3 Lifecycle policies to archive partitions that are older than 30 days to S3 Glacier Flexible R etrieval.**

Athena performance benefits greatly from partition pruning, which reduces scanned data to just the partitions of interest (e.g., last 3 days). Using date prefixes enables efficient queries and minimal I/O. Lifecycle policies meet the 30-day retention requirement automatically. This aligns storage optimization with query performance.

Incorrect answers:

❌ **A. Keep all data without partitioning** – Without partitions, Athena scans far more data than necessary, increasing latency and cost. Manual movement of data is error-prone and operationally heavy. It provides no query-time optimization. It fails both performance and operations goals.

❌ **B. Lambda copying to separate buckets** – Creating many buckets is unnecessary and complicates governance, access, and cataloging. Copying data increases cost and introduces synchronization risks. Lifecycle and partitioning can be achieved within a single bucket. This is more complex than needed.

❌ **D. One bucket per day** – Excessive bucket sprawl is an anti-pattern and complicates permissions, listings, and metadata management. Athena also expects a coherent partitioned layout within a table rather than table-per-bucket. Lifecycle at bucket granularity is inflexible and cumbersome.

Question 16

A company is using Amazon SageMaker to create ML models. The company's data scientists need fine-grained control of the ML workflows that they orchestrate. The data scientists also need the ability to visualize SageMaker jobs and workflows as a directed acyclic graph (DAG). The data scientists must keep a running history of model discovery experiments and must establish model governance for auditing and compliance verifications.

Which solution will meet these requirements?

A. Use AWS CodePipeline and its integration with SageMaker S tudio to manage the entire ML workflows. Use SageMaker ML Lineage T racking for the running history of experiments and for auditing and compliance verifications.

B. Use AWS CodePipeline and its integration with SageMaker Experiments to manage the entire ML workflows. Use SageMaker Experiments for the running history of experiments and for auditing and compliance verifications.

C. Use SageMaker Pipelines and its integration with SageMaker S tudio to manage the entire ML workflows. Use SageMaker ML Lineage T racking for the running history of experiments and for auditing and compliance verifications.

D. Use SageMaker Pipelines and its integration with SageMaker Experiments to manage the entire ML workflows. Use SageMaker Experiments for the running history of experiments and for auditing and compliance verifications.

✅ **Correct answer: D. Use SageMaker Pipelines and its integration with SageMaker Experiments to manage the entire ML workflows. Use SageMaker Experiments for the running history of experiments and for auditing and compliance verifications.**

SageMaker Pipelines provides native DAG-based workflow orchestration and visualization in Studio. SageMaker Experiments tracks trials, parameters, metrics, and artifacts for experiment history and governance. Together they offer

the fine-grained control and visibility requested. This pairing is the intended, fully managed solution for MLOps workflows on SageMaker.

Incorrect answers:

❌ **A. CodePipeline + Lineage Tracking** – CodePipeline is not optimized for ML DAGs or native SageMaker steps, and ML Lineage tracks artifact provenance rather than the full experiment run history and comparisons. This option misses the requested capabilities. It increases integration effort without first-class Pipelines features. It's not the best fit.

❌ **B. CodePipeline + Experiments** – While Experiments is correct for tracking runs, CodePipeline lacks the native ML DAG constructs and visualization that Pipelines provides. You'd have to glue custom actions and lose ML-specific UX. Operational overhead would be higher and features poorer.

❌ **C. Pipelines + Lineage Tracking** – Pipelines is correct for DAGs, but ML Lineage does not replace Experiments for run tracking and governance. You would lose rich experiment comparisons and trial management. This only partially satisfies the tracking requirement.

Question 17

A company needs to give its ML engineers appropriate access to training data. The ML engineers must access training data from only their own business group. The ML engineers must not be allowed to access training data from other business groups.

The company uses a single AWS account and stores all the training data in Amazon S3 buckets. All ML model training occurs in Amazon SageMaker.

Which solution will provide the ML engineers with the appropriate access?

A. Enable S3 bucket versioning.

B. Configure S3 Object Lock settings for each user.

C. Add cross-origin resource sharing (C ORS) policies to the S3 buckets.

D. Create IAM policies. Attach the policies to IAM users or IAM roles.

✅ **Correct answer: D. Create IAM policies. Attach the policies to IAM users or IAM roles.**

Fine-grained, per-group access to S3 data is enforced with IAM policies (and optionally S3 bucket/prefix conditions). You can scope permissions by path prefixes that map to business groups and attach the policies to roles the ML engineers assume. This works cleanly with SageMaker, which uses roles to access training data. It centralizes governance in the account as required.

Incorrect answers:

❌ **A. Enable S3 bucket versioning** – Versioning protects against unintended overwrites and deletions but does not constrain who can access what. It's a data protection feature, not an authorization mechanism. It doesn't address group-based isolation. Access would still be unrestricted without policies.

❌ **B. Configure S3 Object Lock** – Object Lock provides WORM retention and legal hold, not identity-based authorization. It's unrelated to limiting cross-group visibility. Enabling it would complicate data management without solving the access control need. It's not designed for RBAC.

❌ **C. Add CORS policies** – CORS governs browser access from web origins and has nothing to do with IAM-level data scoping for engineers. It does not prevent one group's credentials from listing or reading another group's prefixes. It's orthogonal to the requirement. IAM remains the correct control plane.

Question 18

A company is running ML models on premises by using custom Python scripts and proprietary datasets. The company is using PyT orch. The model building requires unique domain knowledge. The company needs to move the models to AWS.

Which solution will meet these requirements with the LEAST effort?

A. Use SageMaker built-in algorithms to train the proprietary datasets.

B. Use SageMaker script mode and premade images for ML frameworks.

C. Build a container on AWS that includes custom packages and a choice of ML frameworks.

D. Purchase similar production models through AWS Marketplace.

✅ **Correct answer: B. Use SageMaker script mode and premade images for ML frameworks.**

Script mode lets you bring existing PyTorch training code with minimal changes while using AWS-maintained framework images. You don't need to author a custom container unless you have nonstandard system dependencies. This path preserves domain-specific code and accelerates migration. It is typically the least effort for moving custom PyTorch scripts to SageMaker.

Incorrect answers:

❌ **A. Use SageMaker built-in algorithms** – Built-in algorithms are great for common tasks but won't accommodate bespoke PyTorch training logic and domain-specific pipelines. Migrating to a built-in would require extensive rewrites. This contradicts the "least effort" constraint. It sacrifices custom behavior.

❌ **C. Build a custom container** – Custom containers provide maximum flexibility but at higher operational and maintenance cost. For standard PyTorch code, AWS's premade images already cover most dependencies. Building and hardening your own image is unnecessary work here. Reserve this for truly atypical stacks.

❌ **D. Purchase similar models in Marketplace** – Off-the-shelf models won't embed your proprietary dataset nuances or domain tricks. Replacing internal models changes behavior and may not meet accuracy or compliance needs. It also doesn't migrate your codebase. This option doesn't solve the stated problem.

Question 19

An ML engineer receives datasets that contain missing values, duplicates, and extreme outliers. The ML engineer must consolidate these datasets into a single data frame and must prepare the data for ML.

Which solution will meet these requirements?

A. Use Amazon SageMaker Data Wrangler to import the datasets and to consolidate them into a single data frame. Use the cleansing and enrichment functionalities to prepare the data.

B. Use Amazon SageMaker Ground Truth to import the datasets and to consolidate them into a single data frame. Use the human-in-the-loop capability to prepare the data.

C. Manually import and merge the datasets. Consolidate the datasets into a single data frame. Use Amazon Q Developer to generate code snippets that will prepare the data.

D. Manually import and merge the datasets. Consolidate the datasets into the data.

✅ **Correct answer: A. Use Amazon SageMaker Data Wrangler to import the datasets and to consolidate them into a single data frame. Use the cleansing and enrichment functionalities to prepare the data.**

Data Wrangler provides a no/low-code UI with transforms for de-duplication, missing value handling, and outlier treatment. It unifies multiple sources into a single data flow and exports to Pandas/Spark pipelines for production. This minimizes custom coding while ensuring repeatability. It directly addresses consolidation and preparation in one place.

Incorrect answers:

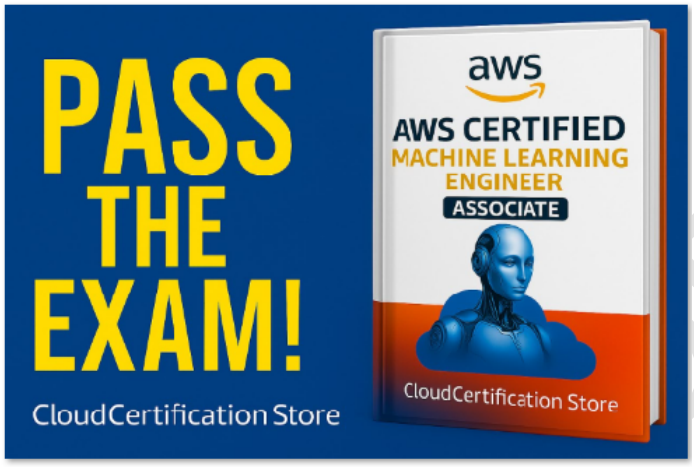
❌ **B. Use SageMaker Ground Truth** – Ground Truth is for data labeling, not for data cleaning, joining, and feature engineering. Human-in-the-loop workflows do

not address deduplication or outlier handling at scale. It's the wrong tool for this job. It increases cost and complexity without benefit.

❌ **C. Manual import + Amazon Q Developer snippets** – While feasible, this relies on ad-hoc code generation and manual stitching. It lacks the integrated preview, profiling, and export capabilities of Data Wrangler. Maintaining the code becomes the team's burden. It's higher effort and less robust.

❌ **D. Manually import and merge; incomplete option** – Even if completed, a purely manual pathway is more error-prone and harder to standardize. It provides no visual lineage of transforms. It doesn't leverage managed transforms or exportable pipelines. It fails the efficiency goal.

(END OF QUESTIONS)



PREVIEW COPY - PLEASE SHARE

Find the full 400+ questions document at

the Cloud Certification Store >> Click Here

All single exams are under $10. You can download free Previews as well.

As a value added bonus, use our courtesy discount code 25-NAY6XE6EEG

Final Review Checklist & Exam Readiness Scorecard

✅ How to Use the Final Review Checklist

This section is meant to **validate your hands-on skills and theoretical readiness** across all exam topics.

Step-by-step:

1. **Print it or load it in a note-taking app** (Notion, Google Docs, OneNote, etc.).
2. Go through each checkbox:
    - ✅ Check it if you **fully understand and can implement** the topic without looking up documentation.
    - ❌ Leave it unchecked if you feel unsure or haven't practiced the task.
3. Prioritize unchecked topics by reviewing:
    - Check the official documentation
    - Practice exams
    - Hands-on labs
4. For each **unchecked item**, write a short action plan or resource link next to it.

📈 How to Use the Exam Readiness Scorecard

This part helps you **self-assess your confidence level** and **focus your revision time** wisely.

Instructions:

1. For each domain (e.g., "Hybrid connectivity and routing"), **rate yourself** from 1 to 5:

   - 1️⃣ = No understanding or hands-on practice
   - 3️⃣ = Moderate familiarity, but need review
   - 5️⃣ = Mastered topic and can apply it in real-world use

2. Add **Notes / Action Items** to explain:
   - Why you scored yourself low
   - What resources you'll use to improve (YouTube, whitepapers, exam guides)
   - Practice test scores if relevant
3. Reassess **2–3 days before your exam**, and compare scores to measure improvement.

🧠 Bonus Tips

- Do **timed mock exams** and cross-reference errors with checklist topics
- Use the scorecard to **simulate an exam debrief**: where did you fail? What must you strengthen?

Once all checklist items are ✅ and all categories are at **4–5 stars** and you're consistently scoring **85%+** on full practice exams with confidence in scenario-based reasoning, then 🎯 you're likely **ready to book the real exam.**

Final Review Checklist

🧠 ML Concepts & Data Preparation

☐ Understand supervised, unsupervised, and reinforcement learning use cases
☐ Identify features, labels, training/test data, and sources of bias
☐ Apply data preprocessing: normalization, missing values, feature engineering
☐ Select correct algorithm for classification, regression, clustering, or recommendation

📦 Data Engineering on AWS

☐ Use S3 as a data lake with lifecycle policies and partitioning
☐ Build data ingestion pipelines with Kinesis, Glue, and Data Pipeline
☐ Use Athena, Glue Data Catalog, and Redshift Spectrum for queries
☐ Apply ETL/ELT transformations in Glue or EMR

⚙️ Model Training & Deployment

☐ Train models using Amazon SageMaker built-in algorithms and custom scripts
☐ Optimize models with hyperparameter tuning jobs and automatic model tuning
☐ Use SageMaker training jobs with managed Spot Training for cost efficiency
☐ Deploy models with SageMaker Endpoints (real-time, batch transform, multi-model)

🔍 Evaluation & Optimization

☐ Select evaluation metrics: accuracy, precision, recall, F1, RMSE, AUC
☐ Detect and address overfitting/underfitting (regularization, dropout, data augmentation)
☐ Apply cross-validation and holdout sets correctly
☐ Optimize inference latency and cost using instance types and autoscaling

🔐 Security & Compliance

☐ Configure IAM roles for SageMaker notebooks, training, and endpoints
☐ Protect data with KMS encryption (at rest & in transit)
☐ Manage secrets with Secrets Manager or Parameter Store
☐ Audit activity with CloudTrail and CloudWatch Logs

📊 Monitoring & Automation

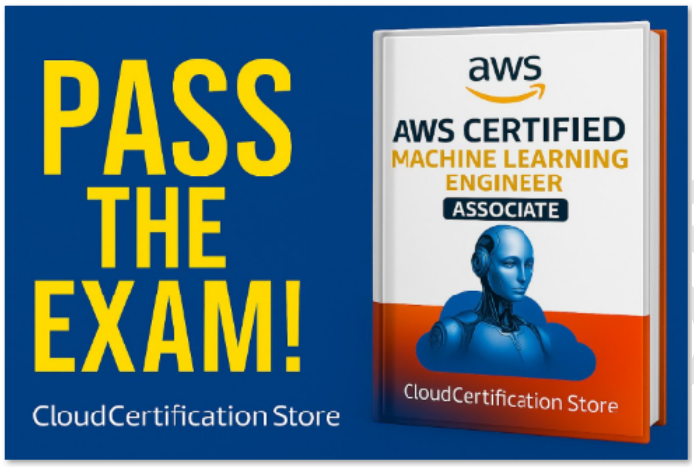
☐ Monitor training and inference jobs with CloudWatch metrics and logs
☐ Enable SageMaker Model Monitor for data drift and bias detection
☐ Automate retraining workflows with Step Functions and EventBridge
☐ Track experiments and lineage with SageMaker Experiments

Exam Readiness Scorecard

Domain	Confidence (1–5)	Notes / Action Items
🧠 ML concepts & data prep	☐1 ☐2 ☐3 ☐4 ☐5	
📦 Data engineering on AWS	☐1 ☐2 ☐3 ☐4 ☐5	
⚙️ Model training & deployment	☐1 ☐2 ☐3 ☐4 ☐5	
🔍 Model evaluation & optimization	☐1 ☐2 ☐3 ☐4 ☐5	
🔐 Security & compliance	☐1 ☐2 ☐3 ☐4 ☐5	
📊 Monitoring, automation & operations	☐1 ☐2 ☐3 ☐4 ☐5	
⏱️ Time management (130-min pacing)	☐1 ☐2 ☐3 ☐4 ☐5	Timed 65-question drill

💫 **Congratulations!! You are on the right path to certification.**

All of our practice exams include **300 + questions**. **This one in particular, contains way over 400 questions.**



PREVIEW COPY - PLEASE SHARE

Find the full 400+ questions document at

the Cloud Certification Store >> Click Here

All single exams are under $10. You can download free Previews as well.

As a value added bonus, use our courtesy discount code 25-NAY6XE6EEG

Our writers who have taken the exam recently—and the reviewers who purchased these materials—agree that **over 90 %** of the questions matched what they saw on the live test.

Invest in your future: browse the full catalogue of Cloud practice exams at our store

Featured Collection

2025 Google Cloud Professional Cloud Architect Exam Questions (GOOG-PCA-0010)

~~$8.89~~ $6.22

2025 Google Cloud Professional Cloud Developer Certification Practice Exam Questions (GOOG-PCD-0010)

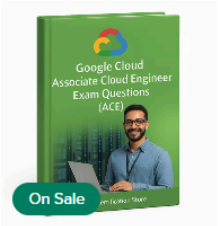
~~$8.89~~ $6.22

2025 Google Cloud Professional Machine Learning Engineer Practice Exam Questions (GOOG-PMLE-0010)

~~$8.99~~ $6.29

2025 Google Cloud Professional Data Engineer Certification Practice Exam Questions (GOOG-PDE-0010)

~~$8.89~~ $6.22

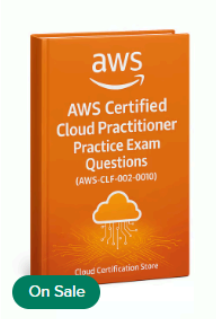


2025 Google Cloud Associate Cloud Engineer Exam Questions (GOOG-ACE-0010)

~~$8.89~~ $6.22

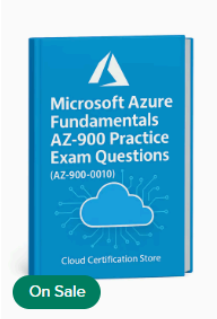
2025 Google Cloud Professional Cloud Database Engineer Practice Exam Questions (GOOG-PCDE-0010)

~~$8.89~~ $6.22



2025 AWS Certified Cloud Practitioner Practice Exam Questions (AWS-CLF-002-0010)

~~$8.99~~ $6.29



2025 Microsoft Azure Fundamentals AZ-900 Practice Exam Questions (AZ-900-0010)

~~$8.89~~ $6.22

2025 Microsoft Azure Developer AZ-204 Practice Exam Questions (AZ-204-0010)

~~$8.89~~ $6.22

FREE Practice Questions for the Google Cloud Generative AI Leader Exam

$0.00+