Download Free Amazon MLA-C01 Exam PDF | PrepBolt

Don't miss out! Download the latest free Amazon AWS Certified Machine Learning Engineer - Associate exam PDF questions. Access real MLA-C01 dumps with verified answers and boost your chances to pass your certification on the first try with PrepBolt MLA-C01 exam pdf questions and answers.

Thank you for Downloading MLA-C01 exam PDF Demo

https://prepbolt.com/MLA-C01.html

Question 1

Question Type: MultipleChoice

A company is building an enterprise AI platform. The company must catalog models for production, manage model versions, and associate metadata such as training metrics with models. The company needs to eliminate the burden of managing different versions of models.

Which solution will meet these requirements?

Options:

A- Use the Amazon SageMaker Model Registry to catalog the models. Create unique tags for each model version. Create key-value pairs to maintain associated metadata.
B- Use the Amazon SageMaker Model Registry to catalog the models. Create model groups for each model to manage the model versions and to maintain associated metadata.
C- Create a separate Amazon Elastic Container Registry (Amazon ECR) repository for each model. Use the repositories to catalog the models and to manage model versions and associated metadata.
D- Create a separate Amazon Elastic Container Registry (Amazon ECR) repository for each model. Create unique tags for each model version. Create key-value pairs to maintain associated metadata.

Answer:

B

Explanation:

AWS enterprise ML best practices recommend using Amazon SageMaker Model Registry to manage models throughout their lifecycle. The Model Registry is designed specifically to catalog models, track versions, and associate metadata such as training metrics, approval status, and deployment history.

Model Registry introduces the concept of model groups, which act as logical containers for different versions of the same model. Each model version within a group automatically inherits versioning, metadata tracking, and governance controls. This eliminates the operational burden of manually managing model versions and ensures consistent lineage and traceability across development, testing, and production environments.

Option A is less optimal because manually tagging model versions increases operational complexity and does not take full advantage of the built-in version management features provided by model groups.

Options C and D are incorrect because Amazon ECR is a container image repository, not a model governance or lifecycle management service. Using ECR to manage ML model versions would require custom tooling and manual metadata handling, significantly increasing operational overhead.

By using model groups within SageMaker Model Registry, the company gains a centralized, scalable, and AWS-native solution for enterprise AI governance. This approach directly aligns with AWS documentation for managing model catalogs, version control, and metadata association while minimizing manual intervention.

Question 2

Question Type: MultipleChoice

A company is running ML models on premises by using custom Python scripts and proprietary datasets. The company is using PyTorch. 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 development effort?

Options:

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

Answer:

B

Explanation:

The company already has custom Python training scripts, proprietary datasets, and uses PyTorch, with significant domain-specific logic embedded in the model. The goal is to migrate these workloads to AWS with the least development effort.

According to AWS documentation, Amazon SageMaker AI script mode is explicitly designed for this scenario. Script mode allows customers to bring their existing training scripts with minimal or no code changes and run them using prebuilt SageMaker framework containers, including PyTorch. This approach eliminates the need to redesign models or rewrite training logic while still benefiting from SageMaker's managed infrastructure, scalability, monitoring, and security.

Option A is incorrect because SageMaker built-in algorithms require adapting data formats and training logic to AWS-provided implementations, which would increase development effort and may not support proprietary domain logic.

Option C is also incorrect because building and maintaining a custom container requires additional

effort for containerization, dependency management, security updates, and lifecycle maintenance---making it more complex than necessary.

Option D is not viable because purchasing models from AWS Marketplace would not support the company's proprietary datasets or unique domain knowledge embedded in existing models.

Therefore, using SageMaker AI script mode with prebuilt PyTorch containers is the fastest, most efficient, and AWS-recommended migration path that minimizes development effort while preserving existing workflows.

Question 3

Question Type: MultipleChoice

A gaming company needs to deploy a natural language processing (NLP) model to moderate a chat forum in a game. The workload experiences heavy usage during evenings and weekends but minimal activity during other hours.

Which solution will meet these requirements MOST cost-effectively?

Options:

A- Use an Amazon SageMaker AI batch transform job with fixed capacity.
B- Use Amazon SageMaker Serverless Inference.
C- Use a single Amazon EC2 GPU instance with reserved capacity.
D- Use Amazon SageMaker Asynchronous Inference.

Answer:

B

Explanation:

The key requirements in this scenario are variable traffic patterns and cost efficiency. The workload has unpredictable spikes during evenings and weekends, followed by long periods of low or no usage. According to AWS Machine Learning documentation, Amazon SageMaker Serverless Inference is specifically designed for such use cases.

SageMaker Serverless Inference automatically provisions, scales, and shuts down compute resources based on incoming inference requests. Customers are billed only for the compute time used during inference, not for idle resources. This makes it highly cost-effective for workloads with intermittent or spiky traffic, such as real-time chat moderation in gaming environments.

Option A is incorrect because batch transform jobs are intended for offline, large-scale inference and require fixed capacity during job execution. They are not suitable for real-time NLP moderation.

Option C is also incorrect because reserving an EC2 GPU instance incurs continuous costs regardless of utilization. This would be inefficient given the long idle periods described in the scenario.

Option D, SageMaker Asynchronous Inference, is designed for workloads with long processing times or large payloads and still requires endpoint provisioning. While it can handle traffic spikes, it does not scale down to zero in the same cost-efficient manner as Serverless Inference.

Therefore, Amazon SageMaker Serverless Inference is the most cost-effective and operationally efficient solution for deploying an NLP moderation model with highly variable usage patterns.

Question 4

Question Type: MultipleChoice

A company runs an ML model on Amazon SageMaker AI. The company uses an automatic process that makes API calls to create training jobs for the model. The company has new compliance rules that prohibit the collection of aggregated metadata from training jobs.

Which solution will prevent SageMaker AI from collecting metadata from the training jobs?

Options:

A- Opt out of metadata tracking for any training job that is submitted.
B- Ensure that training jobs are running in a private subnet in a custom VPC.
C- Encrypt the training data with an AWS Key Management Service (AWS KMS) customer managed key.
D- Reconfigure the training jobs to use only AWS Nitro instances.

Answer:

A

Explanation:

Amazon SageMaker AI automatically collects aggregated metadata from training jobs to improve service reliability, performance, and operational insights. This metadata can include information such as algorithm usage, instance types, resource utilization, and job configuration details. However, AWS documentation clearly states that customers can opt out of SageMaker metadata collection to meet regulatory or compliance requirements.

SageMaker provides a supported mechanism to disable metadata tracking at the training job level. By explicitly opting out of metadata tracking when submitting training jobs---either through the AWS Management Console, AWS CLI, or SDK---the service will stop collecting aggregated metadata for those jobs. This option is specifically designed for customers with strict compliance, data residency, or regulatory constraints.

Option B is incorrect because running training jobs in a private subnet within a custom VPC controls network isolation, not service-level telemetry or metadata collection. Metadata collection occurs at the SageMaker service layer and is independent of VPC configuration.

Option C is also incorrect because encrypting training data with a customer-managed AWS KMS key protects data at rest and in transit but does not prevent SageMaker from collecting operational metadata about training jobs.

Option D is incorrect because AWS Nitro instances provide enhanced security and performance isolation at the infrastructure level but have no impact on SageMaker's metadata collection mechanisms.

Therefore, opting out of metadata tracking for training jobs is the only solution that directly addresses the compliance requirement and is explicitly supported by AWS documentation.

Question 5

Question Type: MultipleChoice

A company is developing a generative AI conversational interface to assist customers with payments. The company wants to use an ML solution to detect customer intent. The company does not have training data to train a model.

Which solution will meet these requirements?

Options:

A- Fine-tune a sequence-to-sequence (seq2seq) algorithm in Amazon SageMaker JumpStart.
B- Use an LLM from Amazon Bedrock with zero-shot learning.
C- Use the Amazon Comprehend DetectEntities API.
D- Run an LLM from Amazon Bedrock on Amazon EC2 instances.

Answer:

B

Explanation:

The key requirement in this scenario is detecting customer intent without having any training data. According to AWS Machine Learning and Generative AI documentation, zero-shot learning is specifically designed for situations where labeled training data is unavailable. Zero-shot learning allows a pre-trained large language model (LLM) to perform tasks it has not been explicitly trained on by leveraging its general knowledge and language understanding.

Amazon Bedrock provides fully managed access to foundation models (FMs) and LLMs that support zero-shot and few-shot learning. By using an LLM from Amazon Bedrock, the company can directly infer customer intent from natural language inputs without building, training, or fine-tuning a custom model. This approach is ideal for conversational interfaces where rapid deployment and scalability are required.

Option A is incorrect because fine-tuning a sequence-to-sequence (seq2seq) model in Amazon SageMaker JumpStart still requires labeled training data. Since the company explicitly does not have training data, this option does not meet the requirement.

Option C is also incorrect because the Amazon Comprehend DetectEntities API is designed for named entity recognition (NER), such as detecting names, dates, locations, or monetary values. It does not perform intent detection and is not suitable for conversational AI intent classification.

Option D is partially misleading. While it is technically possible to run an LLM on Amazon EC2, this does not inherently solve the problem of intent detection without training data. Additionally, Amazon Bedrock already abstracts infrastructure management, scaling, and model hosting, making direct EC2 deployment unnecessary and less efficient.

Therefore, using an LLM from Amazon Bedrock with zero-shot learning is the most appropriate, scalable, and AWS-recommended solution for intent detection without training data.

Thank You for trying MLA-C01 PDF Demo

To try our MLA-C01 practice exam software visit link below

https://prepbolt.com/MLA-C01.html

Start Your MLA-C01 Preparation

Use Coupon “SAVE50” for extra 50% discount on the purchase of Practice Test Software. Test your MLA-C01 preparation with actual exam questions.