Associate Exam Guide (MLA-C01)

AWS Certified Machine Learning Engineer

AWS Certified Machine Learning Engineer: Associate Exam Guide (MLA-C01)

Table of Contents

AWS Certified Machine Learning Engineer - Associate (MLA-C01)

The AWS Certified Machine Learning Engineer - Associate (MLA-C01) exam validates a candidate's ability to build, operationalize, deploy, and maintain machine learning (ML) solutions and pipelines by using the AWS Cloud.

Topics

Introduction

The AWS Certified Machine Learning Engineer - Associate (MLA-C01) exam validates a candidate's ability to build, operationalize, deploy, and maintain machine learning (ML) solutions and pipelines by using the AWS Cloud.

The exam also validates a candidate's ability to complete the following tasks:

- Ingest, transform, validate, and prepare data for ML modeling.

- Select general modeling approaches, train models, tune hyperparameters, analyze model performance, and manage model versions.

- Choose deployment infrastructure and endpoints, provision compute resources, and configure auto scaling based on requirements.

- Set up continuous integration and continuous delivery (CI/CD) pipelines to automate orchestration of ML workflows.

- Monitor models, data, and infrastructure to detect issues.

- Secure ML systems and resources through access controls, compliance features, and best practices.

Target Candidate Description

The target candidate should have at least 1 year of experience using Amazon SageMaker and other AWS services for ML engineering. The target candidate also should have at least 1 year of experience in a related role such as a backend software developer, DevOps developer, data engineer, or data scientist.

Recommended general IT knowledge

The target candidate should have the following general IT knowledge:

- Basic understanding of common ML algorithms and their use cases

- Data engineering fundamentals, including knowledge of common data formats, ingestion, and transformation to work with ML data pipelines

- Knowledge of querying and transforming data

- Knowledge of software engineering best practices for modular, reusable code development, deployment, and debugging

- Familiarity with provisioning and monitoring cloud and on-premises ML resources

- Experience with CI/CD pipelines and infrastructure as code (IaC)

- Experience with code repositories for version control and CI/CD pipelines

Recommended AWS knowledge

The target candidate should have the following AWS knowledge:

- Knowledge of SageMaker capabilities and algorithms for model building and deployment

- Knowledge of AWS data storage and processing services for preparing data for modeling

- Familiarity with deploying applications and infrastructure on AWS

- Knowledge of monitoring tools for logging and troubleshooting ML systems

- Knowledge of AWS services for the automation and orchestration of CI/CD pipelines

- Understanding of AWS security best practices for identity and access management, encryption, and data protection

Job tasks that are out of scope for the target candidate

The following list contains job tasks that the target candidate is not expected to be able to perform. This list is non-exhaustive. These tasks are out of scope for the exam:

- Designing and architecting full end-to-end ML solutions

- Setting up best practices and guiding ML strategies

- Handling integration with a wide array of services or new tools and technologies

- Working deeply in two or more ML domains (for example, natural language processing [NLP], computer vision)

- Quantizing models and analyzing the impact on accuracy

Exam content

Question types

The exam contains one or more of the following question types:

- **Multiple choice:** Has one correct response and three incorrect responses (distractors).
- **Multiple response:** Has two or more correct responses out of five or more response options. You must select all the correct responses to receive credit for the question.
- **Ordering:** Has a list of 3–5 responses to complete a specified task. You must select the correct responses and place the responses in the correct order to receive credit for the question.
- **Matching:** Has a list of responses to match with a list of 3–7 prompts. You must match all the pairs correctly to receive credit for the question.

Unanswered questions on the exam are scored as incorrect. There is no penalty for guessing. The exam includes 50 questions that affect your score.

Unscored content

The exam includes 15 unscored questions that do not affect your score. AWS collects information about performance on these unscored questions to evaluate these questions for future use as scored questions. These unscored questions are not identified on the exam.

Exam results

The AWS Certified Machine Learning Engineer - Associate (MLA-C01) exam has a pass or fail designation. The exam is scored against a minimum standard established by AWS professionals who follow certification industry best practices and guidelines.

Your results for the exam are reported as a scaled score of 100–1,000. The minimum passing score is 720. Your score shows how you performed on the exam as a whole and whether you passed. Scaled scoring models help equate scores across multiple exam forms that might have slightly different difficulty levels.

Your score report could contain a table of classifications of your performance at each section level. The exam uses a compensatory scoring model, which means that you do not need to achieve a passing score in each section. You need to pass only the overall exam.

Each section of the exam has a specific weighting, so some sections have more questions than other sections have. The table of classifications contains general information that highlights your strengths and weaknesses. Use caution when you interpret section-level feedback.

Content outline

This exam guide includes weightings, content domains, and task statements for the exam. This guide does not provide a comprehensive list of the content on the exam. However, additional context for each task statement is available to help you prepare for the exam.

The exam has the following content domains and weightings:

- Content Domain 1: Data Preparation for Machine Learning (ML) (28% of scored content)

- Content Domain 2: ML Model Development (26% of scored content)

- Content Domain 3: Deployment and Orchestration of ML Workflows (22% of scored content)

- Content Domain 4: ML Solution Monitoring, Maintenance, and Security (24% of scored content)

AWS Services for the Exam

The AWS Certified Machine Learning Engineer - Associate exam covers specific AWS services that are relevant to machine learning engineers. Understanding which services are in scope can help you focus your preparation efforts.

For detailed information about the AWS services covered in the exam, see the following section:

- In-Scope AWS Services

Content Domain 1: Data Preparation for Machine Learning (ML)

Tasks

- Task 1.1: Ingest and store data
- Task 1.2: Transform data and perform feature engineering
- Task 1.3: Ensure data integrity and prepare data for modeling

Task 1.1: Ingest and store data

Knowledge of:

- Data formats and ingestion mechanisms (for example, validated and non-validated formats, Apache Parquet, JSON, CSV, Apache ORC, Apache Avro, RecordIO)
- How to use the core AWS data sources (for example, Amazon S3, Amazon Elastic File System [Amazon EFS], Amazon FSx for NetApp ONTAP)
- How to use AWS streaming data sources to ingest data (for example, Amazon Kinesis, Apache Flink, Apache Kafka)
- AWS storage options, including use cases and tradeoffs

Skills in:

- Extracting data from storage (for example, Amazon S3, Amazon Elastic Block Store [Amazon EBS], Amazon EFS, Amazon RDS, Amazon DynamoDB) by using relevant AWS service options (for example, Amazon S3 Transfer Acceleration, Amazon EBS Provisioned IOPS)

- Choosing appropriate data formats (for example, Parquet, JSON, CSV, ORC) based on data access patterns

- Ingesting data into Amazon SageMaker Data Wrangler and SageMaker Feature Store

- Merging data from multiple sources (for example, by using programming techniques, AWS Glue, Apache Spark)

- Troubleshooting and debugging data ingestion and storage issues that involve capacity and scalability

- Making initial storage decisions based on cost, performance, and data structure

Task 1.2: Transform data and perform feature engineering

Knowledge of:

- Data cleaning and transformation techniques (for example, detecting and treating outliers, imputing missing data, combining, deduplication)

- Feature engineering techniques (for example, data scaling and standardization, feature splitting, binning, log transformation, normalization)

- Encoding techniques (for example, one-hot encoding, binary encoding, label encoding, tokenization)

- Tools to explore, visualize, or transform data and features (for example, SageMaker Data Wrangler, AWS Glue, AWS Glue DataBrew)

- Services that transform streaming data (for example, AWS Lambda, Spark)

- Data annotation and labeling services that create high-quality labeled datasets

Skills in:

- Transforming data by using AWS tools (for example, AWS Glue, DataBrew, Spark running on Amazon EMR, SageMaker Data Wrangler)

- Creating and managing features by using AWS tools (for example, SageMaker Feature Store)

- Validating and labeling data by using AWS services (for example, SageMaker Ground Truth, Amazon Mechanical Turk)

Task 1.3: Ensure data integrity and prepare data for modeling

Knowledge of:

- Pre-training bias metrics for numeric, text, and image data (for example, class imbalance [CI], difference in proportions of labels [DPL])
- Strategies to address CI in numeric, text, and image datasets (for example, synthetic data generation, resampling)
- Techniques to encrypt data
- Data classification, anonymization, and masking
- Implications of compliance requirements (for example, personally identifiable information [PII], protected health information [PHI], data residency)

Skills in:

- Validating data quality (for example, by using DataBrew and AWS Glue Data Quality)
- Identifying and mitigating sources of bias in data (for example, selection bias, measurement bias) by using AWS tools (for example, SageMaker Clarify)
- Preparing data to reduce prediction bias (for example, by using dataset splitting, shuffling, and augmentation)
- Configuring data to load into the model training resource (for example, Amazon EFS, Amazon FSx)

Content Domain 2: ML Model Development

Tasks

- [Task 2.1: Choose a modeling approach](#)
- [Task 2.2: Train and refine models](#)
- [Task 2.3: Analyze model performance](#)

Task 2.1: Choose a modeling approach

Knowledge of:

- Capabilities and appropriate uses of ML algorithms to solve business problems

- How to use AWS artificial intelligence (AI) services (for example, Amazon Translate, Amazon Transcribe, Amazon Rekognition, Amazon Bedrock) to solve specific business problems

- How to consider interpretability during model selection or algorithm selection

- Amazon SageMaker AI built-in algorithms and when to apply them

Skills in:

- Assessing available data and problem complexity to determine the feasibility of an ML solution

- Comparing and selecting appropriate ML models or algorithms to solve specific problems

- Choosing built-in algorithms, foundation models, and solution templates (for example, in SageMaker JumpStart and Amazon Bedrock)

- Selecting models or algorithms based on costs

- Selecting AI services to solve common business needs

Task 2.2: Train and refine models

Knowledge of:

- Elements in the training process (for example, epoch, steps, batch size)

- Methods to reduce model training time (for example, early stopping, distributed training)

- Factors that influence model size

- Methods to improve model performance

- Benefits of regularization techniques (for example, dropout, weight decay, L1 and L2)

- Hyperparameter tuning techniques (for example, random search, Bayesian optimization)

- Model hyperparameters and their effects on model performance (for example, number of trees in a tree-based model, number of layers in a neural network)

- Methods to integrate models that were built outside SageMaker AI into SageMaker AI

Skills in:

- Using SageMaker AI built-in algorithms and common ML libraries to develop ML models

- Using SageMaker AI script mode with SageMaker AI supported frameworks to train models (for example, TensorFlow, PyTorch)

- Using custom datasets to fine-tune pre-trained models (for example, Amazon Bedrock, SageMaker JumpStart)
- Performing hyperparameter tuning (for example, by using SageMaker AI automatic model tuning [AMT])
- Integrating automated hyperparameter optimization capabilities
- Preventing model overfitting, underfitting, and catastrophic forgetting (for example, by using regularization techniques, feature selection)
- Combining multiple training models to improve performance (for example, ensembling, stacking, boosting)
- Reducing model size (for example, by altering data types, pruning, updating feature selection, compression)
- Managing model versions for repeatability and audits (for example, by using the SageMaker Model Registry)

Task 2.3: Analyze model performance

Knowledge of:

- Model evaluation techniques and metrics (for example, confusion matrix, heat maps, F1 score, accuracy, precision, recall, Root Mean Square Error [RMSE], receiver operating characteristic [ROC], Area Under the ROC Curve [AUC])
- Methods to create performance baselines
- Methods to identify model overfitting and underfitting
- Metrics available in SageMaker Clarify to gain insights into ML training data and models
- Convergence issues

Skills in:

- Selecting and interpreting evaluation metrics and detecting model bias
- Assessing tradeoffs between model performance, training time, and cost
- Performing reproducible experiments by using AWS services
- Comparing the performance of a shadow variant to the performance of a production variant
- Using SageMaker Clarify to interpret model outputs
- Using SageMaker Model Debugger to debug model convergence

Content Domain 3: Deployment and Orchestration of ML Workflows

Tasks

- Task 3.1: Select deployment infrastructure based on existing architecture and requirements

- Task 3.2: Create and script infrastructure based on existing architecture and requirements

- Task 3.3: Use automated orchestration tools to set up continuous integration and continuous delivery (CI/CD) pipelines

Task 3.1: Select deployment infrastructure based on existing architecture and requirements

Knowledge of:

- Deployment best practices (for example, versioning, rollback strategies)

- AWS deployment services (for example, Amazon SageMaker AI)

- Methods to serve ML models in real time and in batches

- How to provision compute resources in production environments and test environments (for example, CPU, GPU)

- Model and endpoint requirements for deployment endpoints (for example, serverless endpoints, real-time endpoints, asynchronous endpoints, batch inference)

- How to choose appropriate containers (for example, provided or customized)

- Methods to optimize models on edge devices (for example, SageMaker Neo)

Skills in:

- Evaluating performance, cost, and latency tradeoffs

- Choosing the appropriate compute environment for training and inference based on requirements (for example, GPU or CPU specifications, processor family, networking bandwidth)

- Selecting the correct deployment orchestrator (for example, Apache Airflow, SageMaker Pipelines)

- Selecting multi-model or multi-container deployments

- Selecting the correct deployment target (for example, SageMaker AI endpoints, Kubernetes, Amazon Elastic Container Service [Amazon ECS], Amazon Elastic Kubernetes Service [Amazon EKS], AWS Lambda)

- Choosing model deployment strategies (for example, real time, batch)

Task 3.2: Create and script infrastructure based on existing architecture and requirements

Knowledge of:

- Difference between on-demand and provisioned resources

- How to compare scaling policies

- Tradeoffs and use cases of infrastructure as code (IaC) options (for example, AWS CloudFormation, AWS Cloud Development Kit [AWS CDK])

- Containerization concepts and AWS container services

- How to use SageMaker AI endpoint auto scaling policies to meet scalability requirements (for example, based on demand, time)

Skills in:

- Applying best practices to enable maintainable, scalable, and cost-effective ML solutions (for example, automatic scaling on SageMaker AI endpoints, dynamically adding Spot Instances, by using Amazon EC2 instances, by using Lambda behind the endpoints)

- Automating the provisioning of compute resources, including communication between stacks (for example, by using CloudFormation, AWS CDK)

- Building and maintaining containers (for example, Amazon Elastic Container Registry [Amazon ECR], Amazon EKS, Amazon ECS, by using bring your own container [BYOC] with SageMaker AI)

- Configuring SageMaker AI endpoints within the VPC network

- Deploying and hosting models by using the SageMaker AI SDK

- Choosing specific metrics for auto scaling (for example, model latency, CPU utilization, invocations per instance)

Task 3.3: Use automated orchestration tools to set up continuous integration and continuous delivery (CI/CD) pipelines

Knowledge of:

- Capabilities and quotas for AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy
- Automation and integration of data ingestion with orchestration services
- Version control systems and basic usage (for example, Git)
- CI/CD principles and how they fit into ML workflows
- Deployment strategies and rollback actions (for example, blue/green, canary, linear)
- How code repositories and pipelines work together

Skills in:

- Configuring and troubleshooting CodeBuild, CodeDeploy, and CodePipeline, including stages
- Applying continuous deployment flow structures to invoke pipelines (for example, Gitflow, GitHub Flow)
- Using AWS services to automate orchestration (for example, to deploy ML models, automate model building)
- Configuring training and inference jobs (for example, by using Amazon EventBridge rules, SageMaker Pipelines, CodePipeline)
- Creating automated tests in CI/CD pipelines (for example, integration tests, unit tests, end-to-end tests)
- Building and integrating mechanisms to retrain models

Content Domain 4: ML Solution Monitoring, Maintenance, and Security

Tasks

- [Task 4.1: Monitor model inference](#)
- [Task 4.2: Monitor and optimize infrastructure and costs](#)
- [Task 4.3: Secure AWS resources](#)

Task 4.1: Monitor model inference

Knowledge of:

- Drift in ML models
- Techniques to monitor data quality and model performance
- Design principles for ML lenses relevant to monitoring

Skills in:

- Monitoring models in production (for example, by using Amazon SageMaker Model Monitor)
- Monitoring workflows to detect anomalies or errors in data processing or model inference
- Detecting changes in the distribution of data that can affect model performance (for example, by using SageMaker Clarify)
- Monitoring model performance in production by using A/B testing

Task 4.2: Monitor and optimize infrastructure and costs

Knowledge of:

- Key performance metrics for ML infrastructure (for example, utilization, throughput, availability, scalability, fault tolerance)
- Monitoring and observability tools to troubleshoot latency and performance issues (for example, AWS X-Ray, Amazon CloudWatch Lambda Insights, Amazon CloudWatch Logs Insights)
- How to use AWS CloudTrail to log, monitor, and invoke re-training activities
- Differences between instance types and how they affect performance (for example, memory optimized, compute optimized, general purpose, inference optimized)
- Capabilities of cost analysis tools (for example, AWS Cost Explorer, AWS Billing and Cost Management, AWS Trusted Advisor)
- Cost tracking and allocation techniques (for example, resource tagging)

Skills in:

- Configuring and using tools to troubleshoot and analyze resources (for example, CloudWatch Logs, CloudWatch alarms)

- Creating CloudTrail trails

- Setting up dashboards to monitor performance metrics (for example, by using Amazon QuickSight, CloudWatch dashboards)

- Monitoring infrastructure (for example, by using Amazon EventBridge events)

- Rightsizing instance families and sizes (for example, by using SageMaker AI Inference Recommender and AWS Compute Optimizer)

- Monitoring and resolving latency and scaling issues

- Preparing infrastructure for cost monitoring (for example, by applying a tagging strategy)

- Troubleshooting capacity concerns that involve cost and performance (for example, provisioned concurrency, service quotas, auto scaling)

- Optimizing costs and setting cost quotas by using appropriate cost management tools (for example, AWS Cost Explorer, AWS Trusted Advisor, AWS Budgets)

- Optimizing infrastructure costs by selecting purchasing options (for example, Spot Instances, On-Demand Instances, Reserved Instances, SageMaker AI Savings Plans)

Task 4.3: Secure AWS resources

Knowledge of:

- IAM roles, policies, and groups that control access to AWS services (for example, AWS Identity and Access Management [IAM], bucket policies, SageMaker Role Manager)

- SageMaker AI security and compliance features

- Controls for network access to ML resources

- Security best practices for CI/CD pipelines

Skills in:

- Configuring least privilege access to ML artifacts

- Configuring IAM policies and roles for users and applications that interact with ML systems

- Monitoring, auditing, and logging ML systems to ensure continued security and compliance

- Troubleshooting and debugging security issues

- Building VPCs, subnets, and security groups to securely isolate ML systems

In-Scope AWS Services

The following list contains AWS services and features that are in scope for the AWS Certified Machine Learning Engineer - Associate (MLA-C01) exam. This list is non-exhaustive and is subject to change. AWS offerings appear in categories that align with the offerings' primary functions.

Topics

- [Analytics](#)

- [Application Integration](#)

- [Cloud Financial Management](#)

- [Compute](#)

- [Containers](#)

- [Database](#)

- [Developer Tools](#)

- [Machine Learning](#)

- [Management and Governance](#)

- [Media](#)

- [Migration and Transfer](#)

- [Networking and Content Delivery](#)

- [Security, Identity, and Compliance](#)

- [Storage](#)

Analytics

- Amazon Athena

- Amazon Data Firehose

- Amazon EMR

- AWS Glue

- AWS Glue DataBrew

- AWS Glue Data Quality

- Amazon Kinesis

- AWS Lake Formation

- Amazon Managed Service for Apache Flink

- Amazon OpenSearch Service

- Amazon QuickSight

- Amazon Redshift

Application Integration

- Amazon EventBridge

- Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

- Amazon Simple Notification Service (Amazon SNS)

- Amazon Simple Queue Service (Amazon SQS)

- AWS Step Functions

Cloud Financial Management

- AWS Billing and Cost Management

- AWS Budgets

- AWS Cost Explorer

Compute

- AWS Batch

- Amazon EC2

- AWS Lambda

- AWS Serverless Application Repository

Containers

- Amazon Elastic Container Registry (Amazon ECR)

- Amazon Elastic Container Service (Amazon ECS)

- Amazon Elastic Kubernetes Service (Amazon EKS)

Database

- Amazon DocumentDB (with MongoDB compatibility)

- Amazon DynamoDB

- Amazon ElastiCache

- Amazon Neptune

- Amazon RDS

Developer Tools

- AWS Cloud Development Kit (AWS CDK)

- AWS CodeArtifact

- AWS CodeBuild

- AWS CodeDeploy

- AWS CodePipeline

- AWS X-Ray

Machine Learning

- Amazon Augmented AI (Amazon A2I)

- Amazon Bedrock

- Amazon CodeGuru

- Amazon Comprehend

- Amazon Comprehend Medical

- Amazon DevOps Guru

- Amazon Fraud Detector

- AWS HealthLake

- Amazon Kendra

- Amazon Lex

- Amazon Lookout for Equipment

- Amazon Lookout for Metrics

- Amazon Lookout for Vision

- Amazon Mechanical Turk

- Amazon Personalize

- Amazon Polly

- Amazon Q

- Amazon Rekognition

- Amazon SageMaker

- Amazon Textract

- Amazon Transcribe

- Amazon Translate

Management and Governance

- AWS Auto Scaling

- AWS Chatbot

- AWS CloudFormation

- AWS CloudTrail

- Amazon CloudWatch

- Amazon CloudWatch Logs

- AWS Compute Optimizer

- AWS Config

- AWS Organizations

- AWS Service Catalog

- AWS Systems Manager

- AWS Trusted Advisor

Media

- Amazon Kinesis Video Streams

Migration and Transfer

- AWS DataSync

Networking and Content Delivery

- Amazon API Gateway

- Amazon CloudFront

- AWS Direct Connect

- Amazon VPC

Security, Identity, and Compliance

- AWS Identity and Access Management (IAM)

- AWS Key Management Service (AWS KMS)

- Amazon Macie

- AWS Secrets Manager

Storage

- Amazon Elastic Block Store (Amazon EBS)

- Amazon Elastic File System (Amazon EFS)

- Amazon FSx

- Amazon S3

- Amazon S3 Glacier

- AWS Storage Gateway

Out-of-Scope AWS Services

The following list contains AWS services and features that are out of scope for the AWS Certified Machine Learning Engineer - Associate (MLA-C01) exam. This list is non-exhaustive and is subject to change. AWS offerings that are entirely unrelated to the target job roles for the exam are excluded from this list.

Topics

- Analytics

- Application Integration

- Business Applications

- Cloud Financial Management

- Compute

- Containers

- Customer Enablement

- Developer Tools

- End User Computing

- Frontend Web and Mobile

- Internet of Things (IoT)

- Machine Learning

- Management and Governance

- Media

- Migration and Transfer

- Networking and Content Delivery

- Security, Identity, and Compliance

- Storage

Analytics

- AWS Clean Rooms

- Amazon DataZone

- Amazon FinSpace

Application Integration

- Amazon AppFlow

- Amazon MQ

- Amazon Simple Workflow Service (Amazon SWF)

Business Applications

- Amazon Chime

- Amazon Connect

- Amazon Honeycode

- Amazon Pinpoint

- Amazon Simple Email Service (Amazon SES)

- AWS Supply Chain

- AWS Wickr

- Amazon WorkDocs

- Amazon WorkMail

Cloud Financial Management

- AWS Application Cost Profiler

Compute

- AWS App Runner

- AWS Elastic Beanstalk

- Amazon Lightsail

- AWS Outposts

Containers

- Red Hat OpenShift Service on AWS (ROSA)

Customer Enablement

- AWS Activate for startups

- AWS IQ

- AWS re:Post Private

Developer Tools

- AWS Application Composer

- AWS CloudShell

- Amazon CodeCatalyst

- AWS Fault Injection Service

End User Computing

- Amazon AppStream 2.0

- Amazon WorkSpaces

- Amazon WorkSpaces Secure Browser

- Amazon WorkSpaces Thin Client

Frontend Web and Mobile

- AWS Amplify

- AWS AppSync

- AWS Device Farm

- Amazon Location Service

Internet of Things (IoT)

- FreeRTOS

- AWS IoT 1-Click

- AWS IoT Core

- AWS IoT Device Defender

- AWS IoT Device Management

- AWS IoT Events

- AWS IoT FleetWise

- AWS IoT Greengrass

- AWS IoT RoboRunner

- AWS IoT SiteWise
- AWS IoT TwinMaker

Machine Learning

- AWS DeepRacer
- AWS HealthImaging
- AWS HealthOmics
- Amazon Monitron
- AWS Panorama

Management and Governance

- AWS AppConfig
- AWS Control Tower
- AWS Launch Wizard
- AWS License Manager
- Amazon Managed Grafana
- AWS Proton
- AWS Resilience Hub
- AWS Resource Explorer
- AWS Telco Network Builder
- AWS User Notifications

Media

- Amazon Elastic Transcoder
- AWS Elemental Appliances and Software
- AWS Elemental MediaConnect
- AWS Elemental MediaConvert
- AWS Elemental MediaLive
- AWS Elemental MediaPackage

- AWS Elemental MediaStore

- AWS Elemental MediaTailor

- Amazon Interactive Video Service (Amazon IVS)

- Amazon Nimble Studio

Migration and Transfer

- AWS Application Discovery Service

- AWS Application Migration Service

- AWS Mainframe Modernization

- AWS Migration Hub

Networking and Content Delivery

- AWS App Mesh

- AWS Cloud Map

- AWS Global Accelerator

- AWS Private 5G

- Amazon Route 53

- Amazon Route 53 Application Recovery Controller

- Amazon VPC IP Address Manager

Security, Identity, and Compliance

- AWS Artifact

- AWS Audit Manager

- AWS Certificate Manager (ACM)

- AWS CloudHSM

- Amazon Cognito

- Amazon Detective

- AWS Directory Service

- AWS Firewall Manager

- Amazon GuardDuty

- Amazon Inspector

- AWS Payment Cryptography

- AWS Private Certificate Authority

- AWS Resource Access Manager (AWS RAM)

- AWS Security Hub

- AWS Shield

- AWS Signer

- Amazon Verified Permissions

- AWS WAF

Storage

- AWS Elastic Disaster Recovery

Survey

How useful was this exam guide? Let us know by taking our survey.