Exam Guide (DVA-C03)

AWS Certified Developer - Associate

AWS Certified Developer - Associate: Exam Guide (DVA-C03)

Table of Contents

AWS Certified Developer - Associate (DVA-C03)

The AWS Certified Developer - Associate (DVA-C03) exam is intended for individuals who perform a developer role. The exam validates a candidate's ability to demonstrate proficiency in developing, testing, deploying, and debugging AWS Cloud-based applications, including using AI-assisted development tools.

Note: AWS exam guides are periodically reviewed and revised to ensure that each certification exam tests skills and AWS services and features that are current and relevant for the job role(s) that the certification is designed to target. Exam guide revisions will be published at least one month before changes are reflected on your exam. Check the Revisions section for a summary of changes.

Note: This exam is being updated. Registration for the updated version of the exam (DVA-C03) opens December 1, 2026. The last day to take the current version of the exam (DVA-C02) is November 30, 2026.

Topics

Introduction

The AWS Certified Developer - Associate (DVA-C03) exam is intended for individuals who perform a developer role. The exam validates a candidate's ability to demonstrate proficiency in developing, testing, deploying, and debugging AWS Cloud-based applications, including using AI-assisted development tools.

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

- Develop and optimize applications on AWS
- Use AI tools to develop applications
- Package and deploy applications by using continuous integration and continuous delivery (CI/CD) workflows
- Secure application code and data
- Identify and resolve application issues

Target candidate description

The target candidate should have 1 or more years of hands-on experience in developing and maintaining applications by using AWS services.

Recommended general IT knowledge

The target candidate should have the following general IT knowledge:

- Proficiency in at least one high-level programming language
- Understanding of application lifecycle management
- Basic understanding of cloud-focused application development
- Experience in using application development tools, including AI tools
- Familiarity with AI-assisted development workflows

Recommended AWS knowledge

The target candidate should be able to complete the following tasks:

- Develop and secure applications by using AWS service APIs, AWS CLI, and SDKs
- Use a CI/CD pipeline to deploy applications on AWS
- Use AI tools and services to support development, testing, and release of high-quality code
- Integrate managed AI services into applications

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:

- Design architectures (for example, distributed systems, microservices, database schemas and modeling)
- Design and create CI/CD pipelines
- Administer IAM users and groups
- Administer servers and operating systems
- Select foundation models (FMs), engineer AI prompts, and design Retrieval Augmented Generation (RAG) architectures
- Design AI safety frameworks, implement guardrails, and implement AI governance
- Design and create extract, transform, and load (ETL) pipelines
- Design AWS networking infrastructure

This exam guide also includes a list of technologies and concepts that might appear on the exam, a list of in-scope AWS services and features, and a list of out-of-scope AWS services and features.

Exam content

Response types

There are two types of questions on the exam:

- **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.

Unanswered questions 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 Developer - Associate (DVA-C03) 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, tasks, and skills for the exam. This guide does not provide a comprehensive list of the content on the exam.

The exam has the following content domains and weightings:

- Content Domain 1: Development with AWS Services (30% of scored content)
- Content Domain 2: Security (26% of scored content)
- Content Domain 3: Testing and Deployment (22% of scored content)

Content Domain 1: Development with AWS Services

Tasks

Task 1.1: Develop code for applications that are hosted on AWS.

- Skill 1.1.1: Identify tightly coupled and loosely coupled systems, and use AWS services to decouple system components.
- Skill 1.1.2: Choose between synchronous and asynchronous patterns (for example, Amazon API Gateway synchronous invocation, Amazon SQS based asynchronous processing, Amazon EventBridge event-driven workflows).
- Skill 1.1.3: Create and maintain fault-tolerant and resilient applications (for example, by configuring retry with backoff, graceful degradation).
- Skill 1.1.4: Create, extend, and maintain APIs (for example, configure response/request transformations, by enforcing validation rules).
- Skill 1.1.5: Develop code that interacts with AWS services by using APIs and AWS SDKs.
- Skill 1.1.6: Manage streaming data by using AWS services.
- Skill 1.1.7: Serialize and deserialize data to integrate application code with AWS services.
- Skill 1.1.8: Implement event-driven patterns (for example, by using EventBridge, Amazon S3 Event Notifications, Amazon SQS, Amazon SNS).
- Skill 1.1.9: Use AI-assisted development tools to generate, review, and optimize code (for example, spec-driven code generation, automated code reviews, intelligent code completion, refactoring suggestions, security scanning).
- Skill 1.1.10: Integrate managed AWS AI services into applications.

Task 1.2: Develop code for AWS Lambda.

- Skill 1.2.1: Grant Lambda functions access to private resources in VPCs.

- Skill 1.2.2: Configure Lambda functions by defining environment variables and parameters (for example, memory, concurrency, timeout, runtime, handler, layers, extensions).
- Skill 1.2.3: Handle event lifecycles and errors (for example, by using Lambda destinations, dead-letter queues).
- Skill 1.2.4: Integrate Lambda functions with AWS services by using code, event source mappings, and destinations (for example, API Gateway, Amazon DynamoDB).

Task 1.3: Use data stores during application development.

- Skill 1.3.1: Manage data lifecycles (for example, by using Amazon S3 Lifecycle policies, TTL, retention periods).
- Skill 1.3.2: Choose AWS data stores (for example, Amazon ElastiCache, DynamoDB, Amazon S3, Amazon RDS) based on access patterns.
- Skill 1.3.3: Determine the appropriate caching solution based on data store selection.
- Skill 1.3.4: Configure DynamoDB to support applications (for example, by selecting high-cardinality partition keys, applying read consistency models, performing query and scan operations, by defining keys, attributes, and indexes).

Content Domain 2: Security

Tasks

Task 2.1: Implement authentication and authorization mechanisms for applications and AWS services.

- Skill 2.1.1: Implement identity federation.
- Skill 2.1.2: Implement access control (for example, fine-grained permissions, security tokens, MFA).

- Skill 2.1.3: Configure programmatic access to AWS.
- Skill 2.1.4: Use presigned URLs to provide secure, time-limited access (for example, by using Amazon S3 and Amazon CloudFront).
- Skill 2.1.5: Make authenticated calls to AWS services.
- Skill 2.1.6: Define permissions for IAM principals (for example, users, roles, services) and resources.
- Skill 2.1.7: Assume an IAM role.

Task 2.2: Implement encryption and manage sensitive data by using AWS services.

- Skill 2.2.1: Apply encryption at rest and in transit.
- Skill 2.2.2: Manage certificates by using AWS services (for example, AWS Certificate Manager, AWS Private CA).
- Skill 2.2.3: Select and apply client-side or server-side encryption.
- Skill 2.2.4: Use encryption keys to encrypt and decrypt data.
- Skill 2.2.5: Generate certificates and SSH keys for development purposes.
- Skill 2.2.6: Identify, mask, and limit access to sensitive data (for example, personally identifiable information [PII], protected health information [PHI]) by using AWS services.
- Skill 2.2.7: Use secret management services to secure sensitive data.

Task 2.3: Identify and mitigate security risks associated with using and integrating AI services into application development.

- Skill 2.3.1: Request and manage access to AI services.
- Skill 2.3.2: Use data privacy controls to transmit data to AI services (for example, by using VPC endpoints to establish private connectivity, ensuring that inputs and outputs are not used to train AI models).
- Skill 2.3.3: Filter and control AI model inputs and outputs (for example, by applying content filtering, sensitive information detection, PII/PHI redaction, denied topic policies, prompt injection and manipulation protection).
- Skill 2.3.4: Secure AI agent interactions (for example, by using tool-use authorization, session isolation, human-in-the-loop approval flows for sensitive actions).

- Skill 2.3.5: Protect sensitive content in monitoring logs during interactions with AI services.

Content Domain 3: Testing and Deployment

Tasks

Task 3.1: Prepare application artifacts to be deployed to AWS.

- Skill 3.1.1: Manage dependencies for code modules (for example, environment variables, configuration files, container images) within packages.
- Skill 3.1.2: Build and manage container images for deployment (for example, by using Amazon ECR, Docker files, container image tagging, lifecycle policies).
- Skill 3.1.3: Use code repositories in deployment environments.
- Skill 3.1.4: Manage environment-specific application configuration by using AWS services (for example, AWS AppConfig, AWS Systems Manager Parameter Store, AWS Secrets Manager, Amazon API Gateway, AWS Lambda).
- Skill 3.1.5: Use API Gateway features and resources (for example, stage-specific URLs, endpoints).

Task 3.2: Implement and automate application testing in AWS environments.

- Skill 3.2.1: Develop and validate tests (for example, by using AWS SAM, AWS CDK, AWS CloudFormation).
- Skill 3.2.2: Use mock APIs for external dependencies by using AWS services and tools (for example, API Gateway).
- Skill 3.2.3: Create application test events (for example, by using JSON payloads to test Lambda functions and API Gateway APIs).
- Skill 3.2.4: Create and configure application environments (for example, Lambda aliases, container image tags, AWS Amplify branches, AWS Copilot environments, API Gateway APIs).

- Skill 3.2.5: Use AWS AI tools to generate tests and automate repetitive testing workflows (for example, automated test execution, test result analysis, regression testing automation, test coverage).

Task 3.3: Deploy code by using AWS continuous integration and continuous delivery (CI/CD) services.

- Skill 3.3.1: Use Lambda deployment packaging methods to optimize deployment efficiency (for example, .zip files, container images, layers).
- Skill 3.3.2: Deploy containerized applications by using AWS container orchestration services (for example, Amazon ECS task definitions, Amazon EKS manifests, AWS Fargate launch types, rolling updates).
- Skill 3.3.3: Use infrastructure as code (IaC) templates (for example, CloudFormation templates, AWS SAM templates).
- Skill 3.3.4: Deploy application revisions to target environments by using AWS CodeDeploy.
- Skill 3.3.5: Use AWS CodePipeline to orchestrate CI/CD pipelines (for example, invoke pipelines from linked code repositories, orchestrate progression through multiple environments).
- Skill 3.3.6: Implement deployment strategies for application releases (for example, blue/green, canary, rolling).
- Skill 3.3.7: Implement automatic or manual rollback by using AWS services (for example, CloudFormation, CodeDeploy).
- Skill 3.3.8: Use source control features to manage application versions and releases (for example, labels, branches).
- Skill 3.3.9: Use AWS AI tools to support CI/CD workflows (for example, automated deployment approvals, environment provisioning, post-deployment validation).

Content Domain 4: Troubleshooting and Optimization

Tasks

Task 4.1: Assist in a root cause analysis.

- Skill 4.1.1: Debug application code and service integration issues to identify defects.
- Skill 4.1.2: Interpret application metrics, logs, and traces.
- Skill 4.1.3: Query logs to find relevant data.
- Skill 4.1.4: Troubleshoot deployment failures by using output logs.
- Skill 4.1.5: Use AWS AI tools to analyze errors and to generate troubleshooting suggestions.

Task 4.2: Apply logging, monitoring, and observability best practices.

- Skill 4.2.1: Implement effective logging strategies to record application behavior and state (for example, by using structured logging for application events and user actions).
- Skill 4.2.2: Implement code that emits custom metrics.
- Skill 4.2.3: Implement notification alerts for specific actions (for example, quota limits, deployment completions).
- Skill 4.2.4: Implement tracing by using AWS services and tools.
- Skill 4.2.5: Monitor workload health (for example, by using application health checks, liveness and readiness probes).

Task 4.3: Optimize applications by using AWS services and features.

- Skill 4.3.1: Apply concurrency concepts (for example, AWS Lambda provisioned concurrency, Amazon EC2 Auto Scaling, thread pool management, connection pooling).
- Skill 4.3.2: Optimize memory and compute power for an application.
- Skill 4.3.3: Optimize application messaging (for example, by using Amazon SQS queues, Amazon SNS fan-out patterns).
- Skill 4.3.4: Implement caching by using AWS services (for example, Amazon ElastiCache, Amazon DynamoDB Accelerator [DAX], Amazon CloudFront).
- Skill 4.3.5: Detect performance bottlenecks by using AWS services (for example, Amazon CloudWatch).
- Skill 4.3.6: Use AWS AI tools to identify optimization opportunities (for example, performance bottleneck detection, resource usage optimization, code efficiency improvement).

Technologies and Concepts

The following list contains technologies and concepts that might appear on the exam. This list is non-exhaustive and is subject to change. The order and placement of the items in this list is no indication of their relative weight or importance on the exam:

- Analytics
- Application integration
- Compute
- Containers
- Database
- Developer tools
- ML and AI
- Management and governance
- Networking and content delivery
- Security, identity, and compliance
- Storage

Mentions of AWS Services on the Exam

AWS Certification is reducing the reading load on this exam by using official short names of well-known AWS service names that contain abbreviations or parenthetical information. For example, Amazon Simple Notification Service (Amazon SNS) appears on the exam as Amazon SNS.

The Help feature in the exam (available for every question) contains the list of the short AWS service names and their corresponding full names.

You can consult AWS Service Names on the AWS Certification website for the list of services that appear as their short names on the exam. Any services that are on the list but that are out of scope for the exam will not appear on the exam.

Note: Not every abbreviation is fully spelled out on the exam or available in the Help feature. The official full name for some AWS services includes an abbreviation that is never expanded (for example, Amazon API Gateway, Amazon EMR). The exam also might contain other abbreviations that the target audience is expected to know.

In-Scope AWS Services

The following list contains AWS services and features that are in scope for the exam. This list is non-exhaustive and is subject to change. AWS offerings appear in categories that align with the offerings' primary functions:

Analytics

- Amazon Athena
- Amazon Data Firehose
- Amazon Kinesis
- Amazon OpenSearch Service

Application Integration

- Amazon EventBridge
- Amazon SNS
- Amazon SQS
- AWS Step Functions

Compute

- Amazon EC2
- AWS Elastic Beanstalk
- AWS Lambda

Containers

- Amazon ECR
- Amazon ECS
- Amazon EKS

Database

- Amazon Aurora
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon RDS

Developer Tools

- AWS Amplify
- AWS CloudShell
- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline
- AWS X-Ray

Machine Learning and Artificial Intelligence

- Amazon Bedrock
- Amazon Bedrock AgentCore
- Amazon Quick
- Kiro

Management and Governance

- AWS AppConfig
- AWS CDK
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch

- Amazon CloudWatch Logs
- AWS CLI
- AWS SAM
- AWS Systems Manager

Networking and Content Delivery

- Amazon API Gateway
- Amazon CloudFront
- Elastic Load Balancing (ELB)
- AWS PrivateLink
- Amazon Route 53
- Amazon VPC

Security, Identity, and Compliance

- AWS Certificate Manager (ACM)
- Amazon Cognito
- AWS IAM
- AWS KMS
- AWS Private CA
- AWS Secrets Manager
- AWS STS
- AWS WAF

Storage

- Amazon EBS
- Amazon EFS
- Amazon S3
- Amazon S3 Glacier

Out-of-Scope AWS Services

The following list contains AWS services and features that are out of scope for the 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

- Amazon Quick Sight

Business Applications

- Amazon Connect
- Amazon WorkMail

End User Computing

- Amazon AppStream 2.0
- Amazon WorkSpaces

Game Tech

- Amazon GameLift

Machine Learning and Artificial Intelligence

- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker AI
- Strands Agents

Management and Governance

- AWS Managed Services (AMS)

Media

- Amazon Elastic Transcoder

Migration and Transfer

- AWS Application Discovery Service
- AWS Application Migration Service
- AWS Transfer Family

Security, Identity, and Compliance

- AWS Shield Advanced
- AWS Shield Standard

Storage

- AWS Snow Family

- AWS Storage Gateway

Comparison of DVA-C02 and DVA-C03

Side-by-side comparison

The following table shows the domains and the percentage of scored questions in each domain for the DVA-C02 exam (in use until November 30, 2026) and the DVA-C03 exam (in use beginning December 1, 2026).

DVA-C02 domain	DVA-C03 domain
Domain 1: Development with AWS Services (32%)	Domain 1: Development with AWS Services (30%)
Domain 2: Security (26%)	Domain 2: Security (26%)
Domain 3: Deployment (24%)	Domain 3: Testing and Deployment (22%)
Domain 4: Troubleshooting and Optimization (18%)	Domain 4: Troubleshooting and Optimization (22%)

Additions of content for DVA-C03

A new task was introduced in DVA-C03 that addresses five dedicated skills that cover the security aspects of AI service usage.

Task 2.3: Identify and mitigate security risks associated with using and integrating AI services into application development.

- Request and manage access to AI services.
- Use data privacy controls to transmit data to AI services (for example, by using VPC endpoints to establish private connectivity, ensuring inputs and outputs are not used to train AI models).
- Filter and control AI model inputs and outputs (for example, by applying content filtering, sensitive information detection, PII/PHI redaction, denied topic policies, prompt injection and manipulation protection).

- Secure AI agent interactions (for example, by using tool-use authorization, session isolation, human-in-the-loop approval flows for sensitive actions).
- Protect sensitive content in monitoring logs during interactions with AI services.

In Task 1.1, the following content was added:

- Use AI-assisted development tools to generate, review, and optimize code (for example, spec-driven code generation, automated code reviews, intelligent code completion, refactoring suggestions, security scanning).
- Integrate managed AWS AI services into applications.

In Task 2.1, the following content was added:

- Use presigned URLs to provide secure, time-limited access (for example, by using Amazon S3 and Amazon CloudFront)

In Task 3.1, the following content was added:

- Build and manage container images for deployment (for example, by using Amazon ECR, Docker files, container image tagging, lifecycle policies).

In Task 3.2, the following content was added:

- Use AWS AI tools to generate tests and automate repetitive testing workflows (for example, automated test execution, test result analysis, regression testing automation, ensuring test coverage)

In Task 3.3, the following content was added:

- Deploy containerized applications by using AWS container orchestration services (for example, Amazon ECS task definitions, Amazon EKS manifests, AWS Fargate launch types, rolling updates).
- Use AWS AI tools to support CI/CD workflows (for example, automated deployment approvals, environment provisioning, post-deployment validation).

In Task 4.1, the following content was added:

- Use AWS AI tools to analyze errors and to generate troubleshooting suggestions.

In Task 4.3, the following content was added:

- Use AWS AI tools to identify optimization opportunities (for example, performance bottleneck detection, resource usage optimization, code efficiency improvement)

Recategorizations of content for DVA-C03

The following skills and content areas were relocated between domains or tasks. Content retains its scope but is tested in a different location within the exam framework:

What Moved	From (DVA-C02)	To (DVA-C03)
Unit testing by using AWS SAM	Domain 1, Task 1.1	Domain 3, Task 3.2
AWS Lambda function performance tuning	Domain 1, Task 1.2	Domain 4, Task 4.3
Serialize and deserialize data	Domain 1, Task 1.3	Domain 1, Task 1.1
Amazon API Gateway stages and custom domains	Domain 3, Task 3.4	Domain 3, Task 3.1
IaC templates (CDK, AWS SAM, AWS CloudFormation)	Domain 3, Task 3.3	Domain 3, Task 3.3

Structural changes for DVA-C03

DVA-C03 consolidated multiple granular skills into single comprehensive action-oriented skills. These consolidations do not result in a decreased emphasis on essential skills. Rather, the exam has been reorganized to reduce repetition and ensure consistency in level of detail across all domains. The following table shows the before/after for each major consolidation area:

Area	DVA-C02 (Before)	DVA-C03 (After)
Amazon DynamoDB	4 separate skills: Partition keys and design, Consistency models, Query operations and scan operations, Keys and indexing	1 comprehensive DynamoDB skill that covers all aspects
Security (Encryption + Sensitive Data)	Task 2.2 + Task 2.3 = 13 skills total. Task 2.2: Encryption (6 skills), Task 2.3: Sensitive data (7 skills)	Single Task 2.2 = 7 skills (all encryption and data protection in one task)
Testing (Test + Automate)	Task 3.2 + Task 3.3 = 11 skills total. Task 3.2: Test Applications (6 skills), Task 3.3: Automate Testing (5 skills)	Single Task 3.2 = 5 skills (combined with AI testing tool added)
Observability	8 observability skills across Domain 4	5 observability skills (3 merged)
Optimization	9 optimization skills across Domain 4	6 optimization skills (3 merged)

Survey

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