aws training and certification

AWS Certified Developer – Associate (DVA-C01) Exam Guide

Introduction

The AWS Certified Developer – Associate (DVA-C01) exam is intended for individuals who perform a developer role. The exam validates a candidate's ability to do the following:

- Demonstrate an understanding of core AWS services, uses, and basic AWS architecture best practices
- Demonstrate proficiency in developing, deploying, and debugging cloud-based applications by using AWS

Target candidate description

The target candidate should have 1 or more years of hands-on experience developing and maintaining an AWS-based application.

Recommended general IT knowledge

The target candidate should have the following:

- In-depth knowledge of at least one high-level programming language
- Understanding of application lifecycle management
- The ability to write code for serverless applications
- Understanding of the use of containers in the development process

Recommended AWS knowledge

The target candidate should be able to do the following:

- Use the AWS service APIs, CLI, and software development kits (SDKs) to write applications
- Identify key features of AWS services
- Understand the AWS shared responsibility model
- Use a continuous integration and continuous delivery (CI/CD) pipeline to deploy applications on AWS
- Use and interact with AWS services
- Apply basic understanding of cloud-native applications to write code
- Write code by using AWS security best practices (for example, use IAM roles instead of secret and access keys in the code)
- Author, maintain, and debug code modules on AWS

What is considered out of scope for the target candidate?

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

- Design architectures (for example, distributed system, microservices)
- Design and implement CI/CD pipelines

- Administer IAM users and groups
- Administer Amazon Elastic Container Service (Amazon ECS)
- Design AWS networking infrastructure (for example, Amazon VPC, AWS Direct Connect)
- Understand compliance and licensing

For a detailed list of specific tools and technologies that might be covered on the exam, as well as lists of in-scope and out-of-scope AWS services, refer to the Appendix.

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

Select one or more responses that best complete the statement or answer the question. Distractors, or incorrect answers, are response options that a candidate with incomplete knowledge or skill might choose. Distractors are generally plausible responses that match the content area.

Unanswered questions are scored as incorrect; there is no penalty for guessing. The exam includes 50 questions that will affect your score.

Unscored content

The exam includes 15 unscored questions that do not affect your score. AWS collects information about candidate 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-C01) exam is a pass or fail exam. 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. This information is intended to provide general feedback about your exam performance. 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 contains general information that highlights your strengths and weaknesses. Use caution when interpreting section-level feedback.

Content outline

This exam guide includes weightings, test domains, and objectives for the exam. It is not a comprehensive listing of the content on the exam. However, additional context for each of the objectives is available to help guide your preparation for the exam. The following table lists the main content domains and their weightings. The table precedes the complete exam content outline, which includes the additional context. The percentage in each domain represents only scored content.

Domain	% of Exam
Domain 1: Deployment	22%
Domain 2: Security	26%
Domain 3: Development with AWS Services	30%
Domain 4: Refactoring	10%
Domain 5: Monitoring and Troubleshooting	12%
TOTAL	**100%**

Domain 1: Deployment

1.1 Deploy written code in AWS using existing CI/CD pipelines, processes, and patterns.
- Commit code to a repository and invoke build, test and/or deployment actions
- Use labels and branches for version and release management
- Use AWS CodePipeline to orchestrate workflows against different environments
- Apply AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeStar, and AWS CodeDeploy for CI/CD purposes
- Perform a roll back plan based on application deployment policy

1.2 Deploy applications using AWS Elastic Beanstalk.
- Utilize existing supported environments to define a new application stack
- Package the application
- Introduce a new application version into the Elastic Beanstalk environment
- Utilize a deployment policy to deploy an application version (i.e., all at once, rolling, rolling with batch, immutable)
- Validate application health using Elastic Beanstalk dashboard
- Use Amazon CloudWatch Logs to instrument application logging

1.3 Prepare the application deployment package to be deployed to AWS.
- Manage the dependencies of the code module (like environment variables, config files and static image files) within the package
- Outline the package/container directory structure and organize files appropriately
- Translate application resource requirements to AWS infrastructure parameters (e.g., memory, cores)

1.4 Deploy serverless applications.
- Given a use case, implement and launch an AWS Serverless Application Model (AWS SAM) template
- Manage environments in individual AWS services (e.g., Differentiate between Development, Test, and Production in Amazon API Gateway)

Domain 2: Security

2.1 Make authenticated calls to AWS services.
- Communicate required policy based on least privileges required by application.
- Assume an IAM role to access a service
- Use the software development kit (SDK) credential provider on-premises or in the cloud to access AWS services (local credentials vs. instance roles)

2.2 Implement encryption using AWS services.
- Encrypt data at rest (client side; server side; envelope encryption) using AWS services
- Encrypt data in transit

2.3 Implement application authentication and authorization.
- Add user sign-up and sign-in functionality for applications with Amazon Cognito identity or user pools
- Use Amazon Cognito-provided credentials to write code that access AWS services.
- Use Amazon Cognito sync to synchronize user profiles and data
- Use developer-authenticated identities to interact between end user devices, backend authentication, and Amazon Cognito

Domain 3: Development with AWS Services

3.1 Write code for serverless applications.
- Compare and contrast server-based vs. serverless model (e.g., micro services, stateless nature of serverless applications, scaling serverless applications, and decoupling layers of serverless applications)
- Configure AWS Lambda functions by defining environment variables and parameters (e.g., memory, time out, runtime, handler)
- Create an API endpoint using Amazon API Gateway
- Create and test appropriate API actions like GET, POST using the API endpoint
- Apply Amazon DynamoDB concepts (e.g., tables, items, and attributes)
- Compute read/write capacity units for Amazon DynamoDB based on application requirements
- Associate an AWS Lambda function with an AWS event source (e.g., Amazon API Gateway, Amazon CloudWatch event, Amazon S3 events, Amazon Kinesis)
- Invoke an AWS Lambda function synchronously and asynchronously

3.2 Translate functional requirements into application design.
- Determine real-time vs. batch processing for a given use case
- Determine use of synchronous vs. asynchronous for a given use case
- Determine use of event vs. schedule/poll for a given use case
- Account for tradeoffs for consistency models in an application design

3.3 Implement application design into application code.

- Write code to utilize messaging services (e.g., SQS, SNS)
- Use Amazon ElastiCache to create a database cache
- Use Amazon DynamoDB to index objects in Amazon S3
- Write a stateless AWS Lambda function
- Write a web application with stateless web servers (Externalize state)

3.4 Write code that interacts with AWS services by using APIs, SDKs, and AWS CLI.

- Choose the appropriate APIs, software development kits (SDKs), and CLI commands for the code components
- Write resilient code that deals with failures or exceptions (i.e., retries with exponential back off and jitter)

Domain 4: Refactoring

4.1 Optimize applications to best use AWS services and features.

- Implement AWS caching services to optimize performance (e.g., Amazon ElastiCache, Amazon API Gateway cache)
- Apply an Amazon S3 naming scheme for optimal read performance

4.2 Migrate existing application code to run on AWS.

- Isolate dependencies
- Run the application as one or more stateless processes
- Develop in order to enable horizontal scalability
- Externalize state

Domain 5: Monitoring and Troubleshooting

5.1 Write code that can be monitored.

- Create custom Amazon CloudWatch metrics
- Perform logging in a manner available to systems operators
- Instrument application source code to enable tracing in AWS X-Ray

5.2 Perform root cause analysis on faults found in testing or production.

- Interpret the outputs from the logging mechanism in AWS to identify errors in logs
- Check build and testing history in AWS services (e.g., AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline) to identify issues
- Utilize AWS services (e.g., Amazon CloudWatch, VPC Flow Logs, and AWS X-Ray) to locate a specific faulty component

Appendix

Which key tools, technologies, and concepts might be covered on the exam?

The following is a non-exhaustive list of the tools and technologies that could appear on the exam. This list is subject to change and is provided to help you understand the general scope of services, features, or technologies on the exam. The general tools and technologies in this list appear in no particular order. AWS services are grouped according to their primary functions. While some of these technologies will likely be covered more than others on the exam, the order and placement of them in this list is no indication of relative weight or importance:

- Analytics
- Application Integration
- Containers
- Cost and Capacity Management
- Data Movement
- Developer Tools
- Instances (virtual machines)
- Management and Governance
- Networking and Content Delivery
- Security
- Serverless

AWS services and features

Analytics:

- Amazon Elasticsearch Service (Amazon ES)
- Amazon Kinesis

Application Integration:

- Amazon EventBridge (Amazon CloudWatch Events)
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Queue Service (Amazon SQS)
- AWS Step Functions

Compute:

- Amazon EC2
- AWS Elastic Beanstalk
- AWS Lambda

Containers:

- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Services (Amazon EKS)

Database:

- Amazon DynamoDB
- Amazon ElastiCache
- Amazon RDS

Developer Tools:

- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- Amazon CodeGuru
- AWS CodePipeline
- AWS CodeStar
- AWS Fault Injection Simulator
- AWS X-Ray

Management and Governance:

- AWS CloudFormation
- Amazon CloudWatch

Networking and Content Delivery:

- Amazon API Gateway
- Amazon CloudFront
- Elastic Load Balancing

Security, Identity, and Compliance:

- Amazon Cognito
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)

Storage:

- Amazon S3

Out-of-scope AWS services and features

The following is a non-exhaustive list of AWS services and features that are not covered on the exam. These services and features do not represent every AWS offering that is excluded from the exam content. Services or features that are entirely unrelated to the target job roles for the exam are excluded from this list because they are assumed to be irrelevant.

Out-of-scope AWS services and features include the following:

- AWS Application Discovery Service
- Amazon AppStream 2.0
- Amazon Chime
- Amazon Connect
- AWS Database Migration Service (AWS DMS)
- AWS Device Farm
- Amazon Elastic Transcoder
- Amazon GameLift
- Amazon Lex
- Amazon Machine Learning (Amazon ML)
- AWS Managed Services
- Amazon Mobile Analytics
- Amazon Polly

- Amazon QuickSight
- Amazon Rekognition
- AWS Server Migration Service (AWS SMS)
- AWS Service Catalog
- AWS Shield Advanced
- AWS Shield Standard
- AWS Snow Family
- AWS Storage Gateway
- AWS WAF
- Amazon WorkMail
- Amazon WorkSpaces