

Exam Guide (SAA-C03)

AWS Certified Solutions Architect - Associate



AWS Certified Solutions Architect - Associate: Exam Guide (SAA-C03)

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

AWS Certified Solutions Architect - Associate (SAA-C03)	1
Introduction	1
Target candidate description	2
Exam content	2
Response types	2
Unscored content	2
Exam results	2
Content outline	3
Service References	3
Content Domain 1: Design Secure Architectures	3
Task 1.1: Design secure access to AWS resources	4
Task 1.2: Design secure workloads and applications	4
Task 1.3: Determine appropriate data security controls	5
Content Domain 2: Design Resilient Architectures	6
Task 2.1: Design scalable and loosely coupled architectures	6
Task 2.2: Design highly available and/or fault-tolerant architectures	7
Content Domain 3: Design High-Performing Architectures	8
Task 3.1: Determine high-performing and/or scalable storage solutions	8
Task 3.2: Design high-performing and elastic compute solutions	8
Task 3.3: Determine high-performing database solutions	9
Task 3.4: Determine high-performing and/or scalable network architectures	10
Task 3.5: Determine high-performing data ingestion and transformation solutions	10
Content Domain 4: Design Cost-Optimized Architectures	11
Task 4.1: Design cost-optimized storage solutions	11
Task 4.2: Design cost-optimized compute solutions	12
Task 4.3: Design cost-optimized database solutions	13
Task 4.4: Design cost-optimized network architectures	14
Technologies and Concepts	15
Mentions of AWS Services on the Exam	16
In-Scope AWS Services	16
Analytics	17
Application Integration	17
AWS Cost Management	18
Compute	18

Containers	18
Database	18
Developer Tools	19
Front-End Web and Mobile	19
Machine Learning	19
Management and Governance	19
Media Services	20
Migration and Transfer	20
Networking and Content Delivery	21
Security, Identity, and Compliance	21
Serverless	22
Storage	22
Out-of-Scope AWS Services	22
Application Integration	23
AR and VR	23
Blockchain	23
Compute	23
Database	23
Developer Tools	24
Front-End Web and Mobile	24
Game Tech	24
Internet of Things	24
Machine Learning	24
Management and Governance	25
Media Services	25
Migration and Transfer	25
Networking and Content Delivery	25
Quantum Technologies	26
Satellite	26
Survey	26

AWS Certified Solutions Architect - Associate (SAA-C03)

The AWS Certified Solutions Architect - Associate (SAA-C03) exam is intended for individuals who perform a solutions architect role. The exam validates a candidate's ability to design solutions based on the AWS Well-Architected Framework.

Topics

- [Introduction](#)
- [Target candidate description](#)
- [Exam content](#)
- [Content outline](#)
- [Service References](#)
- [Content Domain 1: Design Secure Architectures](#)
- [Content Domain 2: Design Resilient Architectures](#)
- [Content Domain 3: Design High-Performing Architectures](#)
- [Content Domain 4: Design Cost-Optimized Architectures](#)
- [Technologies and Concepts](#)
- [Mentions of AWS Services on the Exam](#)
- [In-Scope AWS Services](#)
- [Out-of-Scope AWS Services](#)
- [Survey](#)

Introduction

The [AWS Certified Solutions Architect - Associate \(SAA-C03\)](#) exam is intended for individuals who perform a solutions architect role. The exam validates a candidate's ability to design solutions based on the AWS Well-Architected Framework.

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

- Design solutions that incorporate AWS services to meet current business requirements and future projected needs
- Design architectures that are secure, resilient, high-performing, and cost-optimized

- Review existing solutions and determine improvements

Target candidate description

The target candidate should have at least 1 year of hands-on experience designing cloud solutions that use AWS services.

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 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 Solutions Architect - Associate (SAA-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, 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: Design Secure Architectures \(30% of scored content\)](#)
- [Content Domain 2: Design Resilient Architectures \(26% of scored content\)](#)
- [Content Domain 3: Design High-Performing Architectures \(24% of scored content\)](#)
- [Content Domain 4: Design Cost-Optimized Architectures \(20% of scored content\)](#)

Service References

The following sections provide detailed information about AWS services, technologies, and concepts relevant to this certification exam:

- [Technologies and Concepts](#)
- [Mentions of AWS Services on the Exam](#)
- [In-Scope AWS Services](#)
- [Out-of-Scope AWS Services](#)

Content Domain 1: Design Secure Architectures

Tasks

- [Task 1.1: Design secure access to AWS resources](#)
- [Task 1.2: Design secure workloads and applications](#)
- [Task 1.3: Determine appropriate data security controls](#)

Task 1.1: Design secure access to AWS resources

Knowledge of:

- Access controls and management across multiple accounts
- AWS federated access and identity services (for example, IAM, AWS IAM Identity Center)
- AWS global infrastructure (for example, Availability Zones, AWS Regions)
- AWS security best practices (for example, the principle of least privilege)
- The AWS shared responsibility model

Skills in:

- Applying AWS security best practices to IAM users and root users (for example, multi-factor authentication [MFA])
- Designing a flexible authorization model that includes IAM users, groups, roles, and policies
- Designing a role-based access control strategy (for example, AWS STS, role switching, cross-account access)
- Designing a security strategy for multiple AWS accounts (for example, AWS Control Tower, service control policies [SCPs])
- Determining the appropriate use of resource policies for AWS services
- Determining when to federate a directory service with IAM roles

Task 1.2: Design secure workloads and applications

Knowledge of:

- Application configuration and credentials security
- AWS service endpoints
- Control ports, protocols, and network traffic on AWS

- Secure application access
- Security services with appropriate use cases (for example, AWS Cognito, AWS GuardDuty, AWS Macie)
- Threat vectors external to AWS (for example, DDoS, SQL injection)

Skills in:

- Designing VPC architectures with security components (for example, security groups, route tables, network ACLs, NAT gateways)
- Determining network segmentation strategies (for example, using public subnets and private subnets)
- Integrating AWS services to secure applications (for example, AWS Shield, AWS WAF, IAM Identity Center, AWS Secrets Manager)
- Securing external network connections to and from the AWS Cloud (for example, VPN, AWS Direct Connect)

Task 1.3: Determine appropriate data security controls

Knowledge of:

- Data access and governance
- Data recovery
- Data retention and classification
- Encryption and appropriate key management

Skills in:

- Aligning AWS technologies to meet compliance requirements
- Encrypting data at rest (for example, AWS KMS)
- Encrypting data in transit (for example, AWS Certificate Manager [ACM] using TLS)
- Implementing access policies for encryption keys
- Implementing data backups and replications
- Implementing policies for data access, lifecycle, and protection
- Rotating encryption keys and renewing certificates

Content Domain 2: Design Resilient Architectures

Tasks

- [Task 2.1: Design scalable and loosely coupled architectures](#)
- [Task 2.2: Design highly available and/or fault-tolerant architectures](#)

Task 2.1: Design scalable and loosely coupled architectures

Knowledge of:

- API creation and management (for example, Amazon API Gateway, REST API)
- AWS managed services with appropriate use cases (for example, AWS Transfer Family, Amazon SQS, AWS Secrets Manager)
- Caching strategies
- Design principles for microservices (for example, stateless workloads compared with stateful workloads)
- Event-driven architectures
- Horizontal scaling and vertical scaling
- How to appropriately use edge accelerators (for example, content delivery network [CDN])
- How to migrate applications into containers
- Load balancing concepts (for example, Application Load Balancer [ALB])
- Multi-tier architectures
- Queuing and messaging concepts (for example, publish/subscribe)
- Serverless technologies and patterns (for example, AWS Fargate, AWS Lambda)
- Storage types with associated characteristics (for example, object, file, block)
- The orchestration of containers (for example, Amazon ECS, Amazon EKS)
- When to use read replicas
- Workflow orchestration (for example, AWS Step Functions)

Skills in:

- Designing event-driven, microservice, and/or multi-tier architectures based on requirements

- Determining scaling strategies for components used in an architecture design
- Determining the AWS services required to achieve loose coupling based on requirements
- Determining when to use containers
- Determining when to use serverless technologies and patterns
- Recommending appropriate compute, storage, networking, and database technologies based on requirements
- Using purpose-built AWS services for workloads

Task 2.2: Design highly available and/or fault-tolerant architectures

Knowledge of:

- AWS global infrastructure (for example, Availability Zones, AWS Regions, Amazon Route 53)
- AWS Managed Services (AMS) with appropriate use cases (for example, Amazon Comprehend, Amazon Polly)
- Basic networking concepts (for example, route tables)
- Disaster recovery (DR) strategies (for example, backup and restore, pilot light, warm standby, active-active failover, recovery point objective [RPO], recovery time objective [RTO])
- Distributed design patterns
- Failover strategies
- Immutable infrastructure
- Load balancing concepts (for example, ALB)
- Proxy concepts (for example, Amazon RDS Proxy)
- Service quotas and throttling (for example, how to configure the service quotas for a workload in a standby environment)
- Storage options and characteristics (for example, durability, replication)
- Workload visibility (for example, AWS X-Ray)

Skills in:

- Determining automation strategies to ensure infrastructure integrity
- Determining the AWS services required to provide a highly available and/or fault-tolerant architecture across AWS Regions or Availability Zones

- Identifying metrics based on business requirements to deliver a highly available solution
- Implementing designs to mitigate single points of failure
- Implementing strategies to ensure the durability and availability of data (for example, backups)
- Selecting an appropriate DR strategy to meet business requirements
- Using AWS services that improve the reliability of legacy applications and applications not built for the cloud (for example, when application changes are not possible)
- Using purpose-built AWS services for workloads

Content Domain 3: Design High-Performing Architectures

Tasks

- [Task 3.1: Determine high-performing and/or scalable storage solutions](#)
- [Task 3.2: Design high-performing and elastic compute solutions](#)
- [Task 3.3: Determine high-performing database solutions](#)
- [Task 3.4: Determine high-performing and/or scalable network architectures](#)
- [Task 3.5: Determine high-performing data ingestion and transformation solutions](#)

Task 3.1: Determine high-performing and/or scalable storage solutions

Knowledge of:

- Hybrid storage solutions to meet business requirements
- Storage services with appropriate use cases (for example, Amazon S3, Amazon EFS, Amazon EBS)
- Storage types with associated characteristics (for example, object, file, block)

Skills in:

- Determining storage services and configurations that meet performance demands
- Determining storage services that can scale to accommodate future needs

Task 3.2: Design high-performing and elastic compute solutions

Knowledge of:

- AWS compute services with appropriate use cases (for example, AWS Batch, Amazon EMR, AWS Fargate)
- Distributed computing concepts supported by AWS global infrastructure and edge services
- Queuing and messaging concepts (for example, publish/subscribe)
- Scalability capabilities with appropriate use cases (for example, Amazon EC2 Auto Scaling, AWS Auto Scaling)
- Serverless technologies and patterns (for example, AWS Lambda, Fargate)
- The orchestration of containers (for example, Amazon ECS, Amazon EKS)

Skills in:

- Decoupling workloads so that components can scale independently
- Identifying metrics and conditions to perform scaling actions
- Selecting the appropriate compute options and features (for example, EC2 instance types) to meet business requirements
- Selecting the appropriate resource type and size (for example, the amount of Lambda memory) to meet business requirements

Task 3.3: Determine high-performing database solutions

Knowledge of:

- AWS global infrastructure (for example, Availability Zones, AWS Regions)
- Caching strategies and services (for example, Amazon ElastiCache)
- Data access patterns (for example, read-intensive compared with write-intensive)
- Database capacity planning (for example, capacity units, instance types, Provisioned IOPS)
- Database connections and proxies
- Database engines with appropriate use cases (for example, heterogeneous migrations, homogeneous migrations)
- Database replication (for example, read replicas)
- Database types and services (for example, serverless, relational compared with non-relational, in-memory)

Skills in:

- Configuring read replicas to meet business requirements
- Designing database architectures
- Determining an appropriate database engine (for example, MySQL compared with PostgreSQL)
- Determining an appropriate database type (for example, Amazon Aurora, Amazon DynamoDB)
- Integrating caching to meet business requirements

Task 3.4: Determine high-performing and/or scalable network architectures

Knowledge of:

- Edge networking services with appropriate use cases (for example, Amazon CloudFront, AWS Global Accelerator)
- How to design network architecture (for example, subnet tiers, routing, IP addressing)
- Load balancing concepts (for example, Application Load Balancer)
- Network connection options (for example, AWS VPN, AWS Direct Connect, AWS PrivateLink)

Skills in:

- Creating a network topology for various architectures (for example, global, hybrid, multi-tier)
- Determining network configurations that can scale to accommodate future needs
- Determining the appropriate placement of resources to meet business requirements
- Selecting the appropriate load balancing strategy

Task 3.5: Determine high-performing data ingestion and transformation solutions

Knowledge of:

- Data analytics and visualization services with appropriate use cases (for example, Amazon Athena, AWS Lake Formation, Amazon QuickSuite)
- Data ingestion patterns (for example, frequency)

- Data transfer services with appropriate use cases (for example, AWS DataSync, AWS Storage Gateway)
- Data transformation services with appropriate use cases (for example, AWS Glue)
- Secure access to ingestion access points
- Sizes and speeds needed to meet business requirements
- Streaming data services with appropriate use cases (for example, Amazon Kinesis)

Skills in:

- Building and securing data lakes
- Designing data streaming architectures
- Designing data transfer solutions
- Implementing visualization strategies
- Selecting appropriate compute options for data processing (for example, Amazon EMR)
- Selecting appropriate configurations for ingestion
- Transforming data between formats (for example, .csv to .parquet)

Content Domain 4: Design Cost-Optimized Architectures

Tasks

- [Task 4.1: Design cost-optimized storage solutions](#)
- [Task 4.2: Design cost-optimized compute solutions](#)
- [Task 4.3: Design cost-optimized database solutions](#)
- [Task 4.4: Design cost-optimized network architectures](#)

Task 4.1: Design cost-optimized storage solutions

Knowledge of:

- Access options (for example, an S3 bucket with Requester Pays object storage)
- AWS cost management service features (for example, cost allocation tags, multi-account billing)
- AWS cost management tools with appropriate use cases (for example, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)

- AWS storage services with appropriate use cases (for example, Amazon FSx, Amazon EFS, Amazon S3, Amazon EBS)
- Backup strategies
- Block storage options (for example, hard disk drive [HDD] volume types, solid state drive [SSD] volume types)
- Data lifecycles
- Hybrid storage options (for example, AWS DataSync, AWS Transfer Family, AWS Storage Gateway)
- Storage access patterns
- Storage tiering (for example, cold tiering for object storage)
- Storage types with associated characteristics (for example, object, file, block)

Skills in:

- Designing appropriate storage strategies (for example, batch uploads to Amazon S3 compared with individual uploads)
- Determining the correct storage size for a workload
- Determining the lowest cost method of transferring data for a workload to AWS storage
- Determining when storage auto scaling is required
- Managing S3 object lifecycles
- Selecting the appropriate backup and/or archival solution
- Selecting the appropriate service for data migration to storage services
- Selecting the appropriate storage tier
- Selecting the correct data lifecycle for storage
- Selecting the most cost-effective storage service for a workload

Task 4.2: Design cost-optimized compute solutions

Knowledge of:

- AWS cost management service features (for example, cost allocation tags, multi-account billing)
- AWS cost management tools with appropriate use cases (for example, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)

- AWS global infrastructure (for example, Availability Zones, AWS Regions)
- AWS purchasing options (for example, Spot Instances, Reserved Instances, Savings Plans)
- Distributed compute strategies (for example, edge processing)
- Hybrid compute options (for example, AWS Outposts)
- Instance types, families, and sizes (for example, memory optimized, compute optimized, virtualization)
- Optimization of compute utilization (for example, containers, serverless computing, microservices)
- Scaling strategies (for example, auto scaling, hibernation)

Skills in:

- Determining an appropriate load balancing strategy (for example, Application Load Balancer [Layer 7] compared with Network Load Balancer [Layer 4] compared with Gateway Load Balancer)
- Determining appropriate scaling methods and strategies for elastic workloads (for example, horizontal compared with vertical, EC2 hibernation)
- Determining cost-effective AWS compute services with appropriate use cases (for example, AWS Lambda, Amazon EC2, AWS Fargate)
- Determining the required availability for different classes of workloads (for example, production workloads, non-production workloads)
- Selecting the appropriate instance family for a workload
- Selecting the appropriate instance size for a workload

Task 4.3: Design cost-optimized database solutions

Knowledge of:

- AWS cost management service features (for example, cost allocation tags, multi-account billing)
- AWS cost management tools with appropriate use cases (for example, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Caching strategies
- Data retention policies

- Database capacity planning (for example, capacity units)
- Database connections and proxies
- Database engines with appropriate use cases (for example, heterogeneous migrations, homogeneous migrations)
- Database replication (for example, read replicas)
- Database types and services (for example, relational compared with non-relational, Amazon Aurora, Amazon DynamoDB)

Skills in:

- Designing appropriate backup and retention policies (for example, snapshot frequency)
- Determining an appropriate database engine (for example, MySQL compared with PostgreSQL)
- Determining cost-effective AWS database services with appropriate use cases (for example, DynamoDB compared with Amazon RDS, serverless)
- Determining cost-effective AWS database types (for example, time series format, columnar format)
- Migrating database schemas and data to different locations and/or different database engines

Task 4.4: Design cost-optimized network architectures

Knowledge of:

- AWS cost management service features (for example, cost allocation tags, multi-account billing)
- AWS cost management tools with appropriate use cases (for example, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Load balancing concepts (for example, Application Load Balancer)
- NAT gateways (for example, NAT instance costs compared with NAT gateway costs)
- Network connectivity (for example, private lines, dedicated lines, VPNs)
- Network routing, topology, and peering (for example, AWS Transit Gateway, VPC peering)
- Network services with appropriate use cases (for example, DNS)

Skills in:

- Configuring appropriate NAT gateway types for a network (for example, a single shared NAT gateway compared with NAT gateways for each Availability Zone)
- Configuring appropriate network connections (for example, AWS Direct Connect compared with VPN compared with internet)
- Configuring appropriate network routes to minimize network transfer costs (for example, Region to Region, Availability Zone to Availability Zone, private to public, AWS Global Accelerator, VPC endpoints)
- Determining strategic needs for content delivery networks (CDNs) and edge caching
- Reviewing existing workloads for network optimizations
- Selecting an appropriate throttling strategy
- Selecting the appropriate bandwidth allocation for a network device (for example, a single VPN compared with multiple VPNs, Direct Connect speed)

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:

- Compute
- Cost management
- Database
- Disaster recovery
- High performance
- Management and governance
- Microservices and component delivery
- Migration and data transfer
- Networking, connectivity, and content delivery
- Resiliency
- Security
- Serverless and event-driven design principles
- Storage

Mentions of AWS Services on the Exam

AWS Certification is reducing the reading load on this exam by using official short names for 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:

Topics

- [Analytics](#)
- [Application Integration](#)
- [AWS Cost Management](#)
- [Compute](#)
- [Containers](#)
- [Database](#)
- [Developer Tools](#)
- [Front-End Web and Mobile](#)
- [Machine Learning](#)
- [Management and Governance](#)

- [Media Services](#)
- [Migration and Transfer](#)
- [Networking and Content Delivery](#)
- [Security, Identity, and Compliance](#)
- [Serverless](#)
- [Storage](#)

Analytics

- Amazon Athena
- AWS Data Exchange
- Amazon Data Firehose
- Amazon EMR
- AWS Glue
- Amazon Kinesis
- AWS Lake Formation
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)
- Amazon OpenSearch Service
- Amazon QuickSuite
- Amazon Redshift

Application Integration

- Amazon AppFlow
- AWS AppSync
- Amazon EventBridge
- Amazon MQ
- Amazon SNS
- Amazon SQS
- AWS Step Functions

AWS Cost Management

- AWS Budgets
- AWS Cost and Usage Report
- AWS Cost Explorer
- Savings Plans

Compute

- AWS Batch
- Amazon EC2
- Amazon EC2 Auto Scaling
- AWS Elastic Beanstalk
- AWS Outposts
- AWS Serverless Application Repository
- VMware Cloud on AWS
- AWS Wavelength

Containers

- Amazon ECR
- Amazon ECS
- Amazon ECS Anywhere
- Amazon EKS
- Amazon EKS Anywhere
- Amazon EKS Distro

Database

- Amazon Aurora
- Amazon Aurora Serverless
- Amazon DocumentDB

- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Keyspaces
- Amazon Neptune
- Amazon RDS
- Amazon Redshift

Developer Tools

- AWS X-Ray

Front-End Web and Mobile

- AWS Amplify
- Amazon API Gateway
- AWS Device Farm

Machine Learning

- Amazon Comprehend
- Amazon Kendra
- Amazon Lex
- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker AI
- Amazon Textract
- Amazon Transcribe
- Amazon Translate

Management and Governance

- AWS Auto Scaling

- AWS CLI
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS Compute Optimizer
- AWS Config
- AWS Control Tower
- AWS Health Dashboard
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Service for Prometheus
- AWS Management Console
- AWS Organizations
- AWS Service Catalog
- AWS Systems Manager
- AWS Trusted Advisor
- AWS Well-Architected Tool

Media Services

- Amazon Elastic Transcoder
- Amazon Kinesis Video Streams

Migration and Transfer

- AWS Application Migration Service
- AWS DataSync
- AWS DMS
- AWS Snow Family
- AWS Transfer Family

Networking and Content Delivery

- AWS Client VPN
- Amazon CloudFront
- AWS Direct Connect
- Elastic Load Balancing (ELB)
- AWS Global Accelerator
- AWS PrivateLink
- Amazon Route 53
- AWS Site-to-Site VPN
- AWS Transit Gateway
- Amazon VPC

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
- AWS IAM Identity Center
- Amazon Inspector
- AWS KMS
- Amazon Macie
- AWS Network Firewall
- AWS Resource Access Manager (AWS RAM)
- AWS Secrets Manager

- AWS Security Hub
- AWS Shield
- AWS WAF
- IAM

Serverless

- AWS AppSync
- AWS Fargate
- AWS Lambda

Storage

- AWS Backup
- Amazon EBS
- Amazon EFS
- Amazon FSx (for all types)
- 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 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

- [Application Integration](#)
- [AR and VR](#)
- [Blockchain](#)
- [Compute](#)

- [Database](#)
- [Developer Tools](#)
- [Front-End Web and Mobile](#)
- [Game Tech](#)
- [Internet of Things](#)
- [Machine Learning](#)
- [Management and Governance](#)
- [Media Services](#)
- [Migration and Transfer](#)
- [Networking and Content Delivery](#)
- [Quantum Technologies](#)
- [Satellite](#)

Application Integration

- Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

AR and VR

- Amazon Sumerian

Blockchain

- Amazon Managed Blockchain

Compute

- Amazon Lightsail

Database

- Amazon RDS on VMware

Developer Tools

- AWS CDK
- AWS CloudShell
- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- Amazon Corretto
- AWS Fault Injection Simulator (AWS FIS)
- AWS Tools and SDKs

Front-End Web and Mobile

- Amazon Location Service

Game Tech

- Amazon GameLift

Internet of Things

- All services

Machine Learning

- Apache MXNet on AWS
- Amazon Augmented AI (Amazon A2I)
- AWS DeepComposer
- AWS Deep Learning AMIs (DLAMI)
- AWS Deep Learning Containers
- Amazon DevOps Guru

- Amazon Elastic Inference
- Amazon HealthLake
- AWS Inferentia
- Amazon Personalize
- PyTorch on AWS
- Amazon SageMaker Canvas
- Amazon SageMaker Ground Truth
- TensorFlow on AWS

Management and Governance

- AWS Console Mobile Application
- AWS Distro for OpenTelemetry

Media Services

- AWS Elemental Appliances and Software
- AWS Elemental MediaConnect
- AWS Elemental MediaConvert
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaTailor
- Amazon Interactive Video Service (Amazon IVS)

Migration and Transfer

- Migration Evaluator

Networking and Content Delivery

- AWS Cloud Map

Quantum Technologies

- Amazon Braket

Satellite

- AWS Ground Station

Survey

How useful was this exam guide? Let us know by [taking our survey](#).