시험 안내서(AIP-C01)

AWS Certified Generative AI Developer - Professional



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

AWS Certified Generative AI Developer - Professional: 시험 안내서(AIP-C01)

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 Generative AI Developer - Professional(AIP-C01) 1
서론 1
대상 응시자 설명 2
권장하는 AWS 지식 2
대상 응시자의 시험 범위에 해당하지 않는 작업 2
시험 콘텐츠 3
문항 유형 3
채점되지 않는 콘텐츠 3
시험 결과 3
내용 개요 4
콘텐츠 도메인 1: 파운데이션 모델 통합, 데이터 관리 및 규정 준수 4
작업 1.1: 요구 사항을 분석하고 GenAI 솔루션 설계 4
작업 1.2: FM 선택 및 구성 4
작업 1.3: FM 사용을 위한 데이터 검증 및 처리 파이프라인 구현 5
작업 1.4: 벡터 저장소 솔루션 설계 및 구현 5
작업 1.5: FM 증강을 위한 검색 메커니즘 설계 6
작업 1.6: FM 상호 작용을 위한 프롬프트 엔지니어링 전략 및 거버넌스 구현 6
콘텐츠 도메인 2: 구현 및 통합 7
작업 2.1: 에이전틱 AI 솔루션 및 도구 통합 구현 7
작업 2.2: 모델 배포 전략 구현 8
작업 2.3: 엔터프라이즈 통합 아키텍처 설계 및 구현 8
작업 2.4: FM API 통합 구현 9
작업 2.5: 애플리케이션 통합 패턴 및 개발 도구 구현 9
콘텐츠 도메인 3: AI 안전, 보안 및 거버넌스 10
작업 3.1: 입력 및 출력 안전 제어 구현 10
작업 3.2: 데이터 보안 및 개인 정보 보호 제어 구현 10
작업 3.3: AI 거버넌스 및 규정 준수 메커니즘 구현 11
작업 3.4: 책임 있는 AI 원칙 구현 11
콘텐츠 도메인 4: GenAI 애플리케이션을 위한 운영 효율성 및 최적화 11
작업 4.1: 비용 최적화 및 리소스 효율성 전략 구현 11
작업 4.2: 애플리케이션 성능 최적화 12
작업 4.3: GenAI 애플리케이션을 위한 모니터링 시스템 구현 12
콘텐츠 도메인 5: 테스트, 검증 및 문제 해결 13
작업 5.1: GenAI용 평가 시스템 구현 13

작업 5.2: GenAI 애플리케이션 문제 해결 14
시험에 출제될 수 있는 기술 및 개념 14
시험에서 언급되는 AWS 서비스 15
범위 내 AWS 서비스 16
분석 16
애플리케이션 통합 17
컴퓨팅 17
컨테이너 17
고객 지원 17
데이터베이스 17
개발자 도구 18
기계 학습 18
AWS의 관리 및 거버넌스 19
마이그레이션 및 전송 20
네트워킹 및 콘텐츠 전송 20
보안, ID 및 규정 준수 20
스토리지 21
범위 외 AWS 서비스 21
애플리케이션 통합 22
분석 22
블록체인 22
비즈니스 애플리케이션 22
클라우드 재무 관리 23
컴퓨팅 23
컨테이너 23
고객 지원 23
데이터베이스 23
개발자 도구 24
최종 사용자 컴퓨팅 24
프런트엔드 웹 및 모바일 24
게임 개발 24
사물 인터넷(IoT) 24
AWS 관리 및 거버넌스 25
기계 학습 25
미디어 서비스 26
마이그레이션 및 전송 26

네트워킹 및 콘텐츠 전송 26
양자 기술 27
로보틱스 27
인공위성 27
설문 조사 27

AWS Certified Generative AI Developer - Professional(AIP-C01)

AWS Certified GenAI Developer - Professional(AIP-C01) 시험은 GenAI 개발자 역할을 수행하는 개인을 대상으로 합니다. 이 시험에서는 응시자가 파운데이션 모델(FM)을 애플리케이션 및 비즈니스 워크플로에 효과적으로 통합하는 능력을 검증합니다. 이 자격증은 AWS 기술을 사용하여 GenAI 솔루션을 프로덕션 환경에 구현하는 방법에 대한 실용적인 지식을 보여줍니다.

주제

- 서론
- 대상 응시자 설명
- 시험 콘텐츠
- 내용 개요
- 콘텐츠 도메인 1: 파운데이션 모델 통합, 데이터 관리 및 규정 준수
- 콘텐츠 도메인 2: 구현 및 통합
- 콘텐츠 도메인 3: AI 안전, 보안 및 거버넌스
- 콘텐츠 도메인 4: GenAI 애플리케이션을 위한 운영 효율성 및 최적화
- 콘텐츠 도메인 5: 테스트, 검증 및 문제 해결
- 시험에 출제될 수 있는 기술 및 개념
- 시험에서 언급되는 AWS 서비스
- 범위 내 AWS 서비스
- 범위 외 AWS 서비스
- 설문 조사

서론

AWS Certified Generative AI Developer - Professional(AIP-C01) 시험은 GenAI 개발자 역할을 수행하는 개인을 대상으로 합니다. 이 시험에서는 응시자가 파운데이션 모델(FM)을 애플리케이션 및 비즈니스 워크플로에 효과적으로 통합하는 능력을 검증합니다. 이 자격증은 AWS 기술을 사용하여 GenAI 솔루션을 프로덕션 환경에 구현하는 방법에 대한 실용적인 지식을 보여줍니다.

또한 이 시험에서는 응시자의 다음 작업을 완료하는 능력을 확인합니다.

- 벡터 저장소, 검색 증강 생성(RAG), 지식 기반 및 기타 GenAI 아키텍처를 사용하여 솔루션을 설계 및 구현
- FM을 애플리케이션 및 비즈니스 워크플로에 통합
- 프롬프트 엔지니어링 및 관리 기술 적용
- 에이전틱 AI 솔루션 구현
- 비용, 성능 및 비즈니스 가치를 위해 GenAI 애플리케이션 최적화
- 보안, 거버넌스, 책임 있는 AI 사례 구현
- GenAI 애플리케이션의 문제 해결, 모니터링 및 최적화
- FM의 품질과 책임 평가

대상 응시자 설명

대상 응시자는 AWS에서 또는 오픈 소스 기술을 사용하여 프로덕션급 애플리케이션을 구축한 경험이 2년 이상이고, 일반 AI/ML 또는 데이터 엔지니어링 경험을 보유하고 있어야 하며, GenAI 솔루션을 구현한 1년 이상의 실무 경험을 보유하고 있어야 합니다.

권장하는 AWS 지식

대상 응시자는 다음과 같은 AWS 지식을 보유하고 있어야 합니다.

- AWS 컴퓨팅, 스토리지 및 네트워킹 서비스 경험
- AWS 보안 모범 사례 및 ID 관리 이해
- AWS 배포 및 코드형 인프라(IaC) 도구 사용 경험
- AWS 모니터링 및 관찰성 서비스에 대한 지식
- AWS 비용 최적화 원칙에 대한 이해

대상 응시자의 시험 범위에 해당하지 않는 작업

다음 목록에는 대상 응시자가 수행할 수 있을 것으로 예상되지 않는 작업이 나와 있습니다. 이 목록에 모든 사항이 포함된 것은 아닙니다. 다음 작업은 시험 범위에 해당하지 않습니다.

- 모델 개발 및 훈련
- 고급 ML 기술

- 데이터 엔지니어링 및 특성 추출

시험 콘텐츠

문항 유형

시험에는 다음 문항 유형 중 1개 이상이 포함됩니다.

- 선다형: 정답 1개와 오답 3개(정답 이외의 답)가 있습니다.
- 복수 응답형: 5개 이상의 응답 중에 2개 이상의 정답이 있습니다. 문항에 배정된 점수를 받으려면 정답을 모두 선택해야 합니다.

답을 하지 않은 문항은 오답으로 채점됩니다. 추측에 따른 불이익은 없습니다. 시험에는 점수에 반영되는 65개의 문항이 포함되어 있습니다.

채점되지 않는 콘텐츠

시험에는 점수에 영향을 주지 않는 10개의 채점되지 않는 문제가 포함됩니다. AWS는 채점되지 않은 문제의 성과에 대한 정보를 수집하여 나중에 채점 문제로 사용할 수 있도록 해당 문항을 평가합니다. 채점되지 않는 문항은 시험에서 식별되지 않습니다.

시험 결과

AWS Certified Generative AI Developer - Professional(AIP-C01) 시험은 합격 또는 불합격이 결정되는 시험입니다. AWS 전문가가 자격증 분야 모범 사례 및 가이드라인에 따라 설정한 최소 표준을 기준으로 시험 점수를 매깁니다.

시험 결과는 100~1,000점의 변환 점수로 보고됩니다. 합격 최소 점수는 750점입니다. 응시자는 점수를 통해 전반적인 시험 성적과 합격 여부를 알 수 있습니다. 변환 점수 모델은 난이도가 조금씩 다를 수 있는 여러 시험 형식에 걸쳐 점수를 균등하게 조정하는 데 도움이 됩니다.

점수 보고서에는 섹션 수준별로 성적 분류표가 포함될 수 있습니다. 시험은 보상 점수 모델을 사용하므로 각 섹션에서 합격 점수를 얻을 필요는 없으며, 전체 시험에만 합격하면 됩니다.

시험의 섹션마다 특정 가중치가 적용되므로 일부 섹션은 다른 섹션보다 문항 수가 많습니다. 분류표에는 응시자의 장단점을 강조하여 보여주는 일반 정보가 포함되어 있습니다. 섹션별 피드백을 파악할 때 주의하시기 바랍니다.

내용 개요

이 시험 안내서에서는 시험의 가중치, 콘텐츠 도메인, 작업 및 기술에 관한 내용을 제공합니다. 이 안내서는 시험 내용의 전체 목록을 제공하지 않습니다.

시험의 콘텐츠 도메인과 가중치는 다음과 같습니다.

- 콘텐츠 도메인 1: 파운데이션 모델 통합, 데이터 관리 및 규정 준수(채점되는 콘텐츠의 31%)
- 콘텐츠 도메인 2: 구현 및 통합(채점되는 콘텐츠의 26%)
- 콘텐츠 도메인 3: AI 안전, 보안 및 거버넌스(채점되는 콘텐츠의 20%)
- 콘텐츠 도메인 4: GenAI 애플리케이션을 위한 운영 효율성 및 최적화(채점되는 콘텐츠의 12%)
- 콘텐츠 도메인 5: 테스트, 검증 및 문제 해결(채점되는 콘텐츠의 11%)

콘텐츠 도메인 1: 파운데이션 모델 통합, 데이터 관리 및 규정 준수

작업 1.1: 요구 사항을 분석하고 GenAI 솔루션 설계

- 기술 1.1.1: 특정 비즈니스 요구 사항 및 기술적 제약에 부합하는 포괄적인 아키텍처 설계 만들기(예: 적절한 FM, 통합 패턴, 배포 전략 사용)
- 기술 1.1.2: 전체 규모로 배포를 진행하기 전에 기술 개념 증명 구현을 개발하여 실현 가능성, 성능 특성 및 비즈니스 가치 검증(예: Amazon Bedrock 사용)
- 기술 1.1.3: 표준화된 기술 구성 요소를 만들어 여러 배포 시나리오에서 일관된 구현 보장(예: AWS Well-Architected Framework, AWS WA Tool Generative AI Lens 사용)

작업 1.2: FM 선택 및 구성

- 기술 1.2.1: FM을 평가하고 선택하여 특정 비즈니스 사용 사례 및 기술 요구 사항에 맞게 최적으로 조정(예: 성능 벤치마크, 역량 분석, 한계 평가 사용)
- 기술 1.2.2: 코드 수정 없이 동적 모델 선택 및 기관 전환이 가능하도록 유연한 아키텍처 패턴 만들기(예: AWS Lambda, Amazon API Gateway, AWS AppConfig 사용)
- 기술 1.2.3: 서비스 중단 중에도 지속적으로 운영할 수 있는 복원력을 갖춘 AI 시스템 설계(예: AWS Step Functions 서킷 브레이커 패턴, 리전별 가용성이 제한된 모델에 대한 Amazon Bedrock 크로스 리전 추론, 크로스 리전 모델 배포, 적절한 성능 저하 전략 사용)
- 기술 1.2.4: FM 사용자 지정 배포 및 수명 주기 관리 구현(예: 도메인별 미세 조정된 모델 배포에 Amazon SageMaker AI 사용, 모델 배포를 위한 저랭크 적응(LoRA) 및 어댑터와 같은 파라미터 효율

적인 조정 기술, 버전 관리 및 사용자 지정 모델 배포를 위한 SageMaker 모델 레지스트리, 모델 업데이트를 위한 자동 배포 파이프라인, 실패한 배포에 대한 롤백 전략, 모델 사용 중지 및 교체를 위한 수명 주기 관리 사용)

작업 1.3: FM 사용을 위한 데이터 검증 및 처리 파이프라인 구현

- 기술 1.3.1: 포괄적인 데이터 검증 워크플로를 만들어 데이터가 FM 사용에 대한 품질 표준을 충족하는지 확인(예: AWS Glue Data Quality, SageMaker Data Wrangler, 사용자 지정 Lambda 함수, Amazon CloudWatch 지표 사용)
- 기술 1.3.2: FM 사용을 위한 특수 처리 요구 사항을 통해 텍스트, 이미지, 오디오 및 표 형식 데이터를 비롯한 복잡한 데이터 유형을 처리하는 데이터 처리 워크플로 생성(예: Amazon Bedrock 멀티모달 모델, SageMaker Processing, AWS Transcribe, 고급 멀티모달 파이프라인 아키텍처 사용)
- 기술 1.3.3: 모델별 요구 사항에 따라 FM 추론을 위한 입력 데이터의 형식 지정(예: Amazon Bedrock API 요청에 JSON 형식 지정 사용, SageMaker AI 엔드포인트를 위한 구조화된 데이터 준비, 대화 기반 애플리케이션을 위한 대화 형식 지정)
- 기술 1.3.4: 입력 데이터 품질을 개선하여 FM 응답 품질 및 일관성 개선(예: Amazon Bedrock을 사용하여 텍스트 다시 지정, Amazon Comprehend를 사용하여 항목 추출, Lambda 함수를 사용하여 데이터 정규화)

작업 1.4: 벡터 저장소 솔루션 설계 및 구현

- 기술 1.4.1: FM 증강을 위한 고급 벡터 데이터베이스 아키텍처를 만들어 기존 검색 기능을 넘어서는 효율적인 시맨틱 검색 사용(예: 계층적 구성을 위한 Amazon Bedrock Knowledge Base, 주제 기반 세분화로 Amazon Bedrock 통합을 위한 Neural 플러그 인이 포함된 Amazon OpenSearch Service, Amazon S3 문서 리포지토리를 사용하는 Amazon RDS, 메타데이터 및 임베딩을 위한 벡터 데이터베이스를 갖춘 Amazon DynamoDB 사용)
- 기술 1.4.2: 포괄적인 메타데이터 프레임워크를 개발하여 FM 상호 작용에 대한 검색 정밀도와 컨텍스트 인식 개선(예: 문서 타임스탬프용 S3 객체 메타데이터, 소유권 정보에 대한 사용자 지정 속성, 도메인 분류를 위한 태깅 시스템 사용)
- 기술 1.4.3: 고성능 벡터 데이터베이스 아키텍처를 구현하여 FM 검색을 위한 대규모 시맨틱 검색 성능 최적화(예: OpenSearch 샤딩 전략, 특수 도메인에 대한 다중 인덱스 접근 방식, 계층적 인덱싱 기술 사용)
- 기술 1.4.4: AWS 서비스를 사용하여 리소스와 연결하는 통합 구성 요소 만들기(예: 문서 관리 시스템, 지식 기반, GenAI 애플리케이션의 포괄적인 데이터 통합을 위한 내부 위키)

- 기술 1.4.5: 벡터 저장소에 FM 증강을 위한 정확한 최신 정보가 포함되도록 데이터 유지 관리 시스템 설계 및 배포(예: 증분의의 업데이트 메커니즘, 실시간 변경 감지 시스템, 자동화된 동기화 워크플로, 예약된 새로 고침 파이프라인 사용)

작업 1.5: FM 증강을 위한 검색 메커니즘 설계

- 기술 1.5.1: 효과적인 문서 분할 접근 방식을 개발하여 FM 컨텍스트 증대를 위한 검색 성능 최적화(예: Amazon Bedrock 청킹 기능, 고정 크기 청킹 구현에 Lambda 함수 사용, 콘텐츠 구조 기반 계층적 청킹에 대한 사용자 지정 처리)
- 기술 1.5.2: 시맨틱 검색에 효율적인 벡터 표현을 만들기 위한 최적의 임베딩 솔루션 선택 및 구성(예: 차원 및 도메인 적합성을 기반으로 하는 Amazon Titan 임베딩 사용, Amazon Bedrock 임베딩 모델의 성능 특성 평가, Lambda 함수를 사용하여 임베딩 일괄 만들기)
- 기술 1.5.3: FM 증강을 위한 시맨틱 검색 기능을 사용하도록 벡터 검색 솔루션 배포 및 구성(예: 벡터 검색 기능을 갖춘 OpenSearch Service, pgvector 확장을 지원하는 Amazon Aurora, 관리형 벡터 스토어 기능을 갖춘 Amazon Bedrock Knowledge Base 사용)
- 기술 1.5.4: FM 컨텍스트에서 검색된 정보의 관련성과 정확성을 개선하는 고급 검색 아키텍처 만들기(예: 시맨틱 검색을 위한 OpenSearch 사용, 키워드와 벡터를 결합하는 하이브리드 검색, Amazon Bedrock 리랭커 모델 사용)
- 기술 1.5.5: 정교한 쿼리 처리 시스템을 개발하여 FM 확장의 검색 효율성 및 결과 품질 개선(예: 쿼리 확장에 Amazon Bedrock, 쿼리 분해에 Lambda 함수, 쿼리 변환에 Step Functions 사용)
- 기술 1.5.6: FM과 원활하게 통합할 수 있는 일관된 액세스 메커니즘 만들기(예: 벡터 검색을 위한 함수 호출 인터페이스, 벡터 쿼리를 위한 모델 컨텍스트 프로토콜(MCP) 클라이언트, 검색 증대를 위한 표준화된 API 패턴 사용)

작업 1.6: FM 상호 작용을 위한 프롬프트 엔지니어링 전략 및 거버넌스 구현

- 기술 1.6.1: FM 동작 및 출력을 제어하는 효과적인 모델 지침 프레임워크 만들기(예: Amazon Bedrock Prompt Management를 사용하여 역할 정의 적용, Amazon Bedrock Guardrails를 사용하여 책임 있는 AI 지침 적용, 응답 형식을 지정하는 템플릿 구성 사용)
- 기술 1.6.2: 대화형 AI 시스템을 구축하여 컨텍스트를 유지하고 FM과의 사용자 상호 작용 개선(예: 설명 워크플로에 Step Functions, 의도 인식에 Amazon Comprehend, 대화 기록 저장에 DynamoDB 사용)
- 기술 1.6.3: 포괄적인 프롬프트 관리 및 거버넌스 시스템을 구현하여 FM 운영의 일관성 및 감독 보장(예: Amazon Bedrock Prompt Management를 사용하여 파라미터화된 템플릿 및 승인 워크플로

만들기, Amazon S3를 사용하여 템플릿 리포지토리 저장, AWS CloudTrail을 사용하여 사용 추적, Amazon CloudWatch Logs를 사용하여 액세스 기록)

- 기술 1.6.4: FM에 즉각적인 효과와 신뢰성을 보장하는 품질 보증 시스템 개발(예: Lambda 함수를 사용하여 예상 출력 확인, Step Functions를 사용하여 엣지 케이스 테스트, CloudWatch를 사용하여 프롬프트 회귀 테스트)
- 기술 1.6.5: FM 성능을 개선하여 프롬프트를 반복적으로 미세 조정하고 기본 프롬프트 기술을 넘어서는 응답 품질 개선(예: 구조화된 입력 구성 요소, 출력 형식 사양, 생각의 사슬 명령 패턴, 피드백 루프 사용)
- 기술 1.6.6: FM을 사용하여 정교한 작업을 처리할 수 있는 복잡한 프롬프트 시스템 설계(예: 순차적 프롬프트 체인에 Amazon Bedrock Prompt Flows 사용, 모델 응답을 기반으로 하는 조건부 분기, 재사용 가능한 프롬프트 구성 요소, 통합형 전처리 및 후처리 단계 사용)

콘텐츠 도메인 2: 구현 및 통합

작업 2.1: 에이전틱 AI 솔루션 및 도구 통합 구현

- 기술 2.1.1: 적절한 메모리 및 상태 관리 기능을 갖춘 지능형 자율 시스템 개발(예: 다중 에이전트 시스템에 Strands 에이전트 및 AWS Agent Squad, 에이전트-도구 상호 작용에 MCP 사용)
- 기술 2.1.2: FM이 구조화된 추론 단계에 따라 복잡한 문제를 분류하고 해결하는 고급 문제 해결 시스템 만들기(예: Step Functions를 사용하여 ReAct 패턴 및 생각의 사슬 추론 접근법 구현)
- 기술 2.1.3: 안전한 AI 워크플로를 개발하여 제어된 FM 동작 보장(예: 중지 조건을 구현하기 위한 Step Functions, 타임아웃 메커니즘을 구현하는 Lambda 함수, 리소스 경계를 적용하는 IAM 정책, 실패 완화를 위한 서킷 브레이커 사용)
- 기술 2.1.4: 여러 기능에서 성능을 최적화하는 정교한 모델 조정 시스템 만들기(예: 특수 FM을 사용하여 복잡한 작업 수행, 모델 앙상블에 대한 사용자 지정 집계 로직, 모델 선택 프레임워크 수행)
- 기술 2.1.5: 인간의 전문 지식으로 FM 기능을 향상하는 협업 AI 시스템 개발(예: 검토 및 승인 프로세스 조정에 Step Functions 사용, 피드백 수집 메커니즘 구현에 API Gateway 사용, 인간 증강 패턴 사용)
- 기술 2.1.6: 지능형 도구 통합을 구현하여 FM 기능을 확장하고 신뢰할 수 있는 도구 운영 보장(예: 사용자 지정 동작 구현에 Strand API 사용, 표준화된 함수 정의 사용, 오류 처리 및 파라미터 검증을 구현하는 Lambda 함수 사용)
- 기술 2.1.7: 모델 확장 프레임워크를 개발하여 FM 기능 개선(예: 경량 도구 액세스를 제공하는 스테이트리스 MCP 서버 구현에 Lambda 함수 사용, 복잡한 도구를 제공하는 MCP 서버 구현에 Amazon ECS 사용, 일관된 액세스 패턴 보장에 MCP 클라이언트 라이브러리 사용)

작업 2.2: 모델 배포 전략 구현

- 기술 2.2.1: 특정 애플리케이션 요구 사항 및 성능 요구 사항을 기반으로 FM 배포(예: 온디맨드 호출에 Lambda 함수 사용, Amazon Bedrock 프로비저닝 처리량 구성 사용, 하이브리드 솔루션 구현에 SageMaker AI 엔드포인트 사용)
- 기술 2.2.2: 기존 ML 배포와는 다른 대규모 언어 모델(LLM)의 고유한 문제를 해결하여 FM 솔루션 배포(예: 특수 모델 로딩 전략에 따라 메모리 요구 사항 구현, GPU 사용률 구현, 토큰 처리 용량에 최적화된 컨테이너 기반 배포 패턴 구현)
- 기술 2.2.3: GenAI 워크로드에 대한 성능과 리소스 요구 사항의 균형을 맞추는 최적화된 FM 배포 접근 방식 개발(예: 적절한 모델 선택, 특정 작업에 대해 사전 학습된 소규모 모델 사용, API 기반 모델 캐스케이딩을 사용하여 일상적인 쿼리 수행)

작업 2.3: 엔터프라이즈 통합 아키텍처 설계 및 구현

- 기술 2.3.1: 기존 엔터프라이즈 환경에 FM 기능을 원활하게 통합하는 엔터프라이즈 연결 솔루션 만들기(예: 레거시 시스템과의 API 기반 통합 사용, 느슨한 결합에 이벤트 기반 아키텍처 사용, 데이터 동기화 패턴 구현)
- 기술 2.3.2: 통합 AI 기능을 개발하여 GenAI 기능으로 기존 애플리케이션 개선(예: 마이크로서비스 통합 구현에 API Gateway 사용, 웹후크 핸들러를 위한 Lambda 함수 사용, 이벤트 기반 통합 구현에 Amazon EventBridge 사용)
- 기술 2.3.3: 적절한 보안 제어를 보장하는 보안 액세스 프레임워크 만들기(예: FM 서비스와 엔터프라이즈 시스템 간의 ID 페더레이션 사용, 모델 및 데이터 액세스에 역할 기반 액세스 제어 사용, FM에 최소 권한 API 액세스 사용)
- 기술 2.3.4: FM 액세스를 지원하는 동시에 관할 구역 전반의 데이터 규정 준수를 보장하는 환경 간 AI 솔루션 개발(예: 온프레미스 데이터 통합에 AWS Outposts 사용, 엣지 배포 수행에 AWS Wavelength 사용, 클라우드와 온프레미스 리소스 간 보안 라우팅 사용)
- 기술 2.3.5: CI/CD 파이프라인 및 GenAI 게이트웨이 아키텍처를 구현하여 엔터프라이즈 환경에서 안전하고 규정을 준수하는 소비 패턴 구현(예: AWS CodePipeline, AWS CodeBuild, 보안 스캔 및 롤백 지원이 포함된 GenAI 구성 요소의 지속적 배포 및 테스트를 위한 자동화된 테스트 프레임워크, 중앙 집중식 추상화 계층, 관찰성 및 제어 메커니즘 사용)

작업 2.4: FM API 통합 구현

- 기술 2.4.1: 유연한 모델 상호 작용 시스템 만들기(예: 다양한 컴퓨팅 환경의 동기 요청 관리에 Amazon Bedrock API 사용, 언어별 비동기 처리 수행에 AWS SDK 및 Amazon SQS 사용, 사용자 지정 API 클라이언트에 요청 검증을 제공하기 위해 API Gateway 사용)
- 기술 2.4.2: FM으로부터 즉각적인 피드백을 제공하는 실시간 AI 상호 작용 시스템 개발(예: 증분 응답 전송에 Amazon Bedrock 스트리밍 API 사용, 텍스트 실시간 생성에 WebSockets 또는 서버 전송 이벤트 사용, 청크 전송 인코딩 구현에 API Gateway 사용)
- 기술 2.4.3: 신뢰할 수 있는 운영을 보장하는 복원력이 뛰어난 FM 시스템 만들기(예: 지수 백오프에 AWS SDK, 속도 제한 관리에 API Gateway, 정상적인 성능 저하에 폴백 메커니즘, 서비스 경계 전반에서 관찰성 제공에 AWS X-Ray 사용)
- 기술 2.4.4: 지능형 모델 라우팅 시스템을 개발하여 모델 선택 최적화(예: 정적 라우팅 구성 구현에 애플리케이션 코드 사용, 동적 콘텐츠 기반 특수 FM에 대한 라우팅에 Step Functions 사용, 지표 기반 지능형 모델 라우팅 사용, 라우팅 로직에 대한 요청 변환이 포함된 API Gateway 사용)

작업 2.5: 애플리케이션 통합 패턴 및 개발 도구 구현

- 기술 2.5.1: GenAI 워크로드의 특정 요구 사항을 해결하는 FM API 인터페이스 만들기(예: 스트리밍 응답 처리에 API Gateway 사용, 토큰 제한 관리 사용, 모델 타임아웃 처리를 위한 재시도 전략 사용)
- 기술 2.5.2: 액세스 가능한 AI 인터페이스를 개발하여 FM의 도입 및 통합 가속화(예: 선언형 UI 구성 요소 개발에 AWS Amplify 사용, API 우선 개발 접근 방식을 위한 OpenAPI 사양 개발, 노코드 워크플로 빌더를 위한 Amazon Bedrock Prompt Flows 사용)
- 기술 2.5.3: 비즈니스 시스템 개선 사항 만들기(예: 고객 관계 관리(CRM) 개선 사항 구현에 Lambda 함수 사용, 문서 처리 시스템 오케스트레이션에 Step Functions 사용, 자동화된 데이터 처리 워크플로 관리에 Amazon Bedrock Data Automation 사용)
- 기술 2.5.4: 개발자 생산성을 향상하여 GenAI 애플리케이션의 개발 워크플로 가속화(예: 코드 생성 및 리팩터링에 Amazon Q Developer 사용, API 지원을 위한 코드 제안, AI 구성 요소 테스트, 성능 최적화)
- 기술 2.5.5: 고급 GenAI 애플리케이션을 개발하여 정교한 AI 기능 구현(예: AWS 네이티브 오케스트레이션을 위한 Strands Agents 및 AWS Agent Squad 사용, 에이전트 설계 패턴 오케스트레이션용 Step Functions 사용, 프롬프트 체인 패턴 관리를 위한 Amazon Bedrock 사용)
- 기술 2.5.6: FM 애플리케이션의 문제 해결 효율성 개선(예: 프롬프트와 응답 분석에 CloudWatch 로그 인사이트 사용, FM API 호출 추적에 X-Ray 사용, GenAI 관련 오류 패턴 인식 구현에 Amazon Q Developer 사용)

콘텐츠 도메인 3: AI 안전, 보안 및 거버넌스

작업 3.1: 입력 및 출력 안전 제어 구현

- 기술 3.1.1: FM에 대한 유해한 사용자 입력으로부터 보호하기 위한 포괄적인 콘텐츠 안전 시스템 개발(예: 콘텐츠 필터링에 Amazon Bedrock 가드레일 사용, 사용자 지정 조정 워크플로 구현에 Step Functions 및 Lambda 함수 사용, 실시간 검증 메커니즘 구현)
- 기술 3.1.2: 유해한 출력을 방지하는 콘텐츠 안전 프레임워크 만들기(예: Amazon Bedrock 가드레일을 사용하여 응답 필터링, 콘텐츠 조정 및 유해성 탐지를 위한 특수 FM 평가, 결정적 결과를 보장하기 위한 텍스트-SQL 변환)
- 기술 3.1.3: FM 응답에서 할루시네이션을 줄이기 위한 정확도 검증 시스템 개발(예: Amazon Bedrock Knowledge Base를 사용하여 응답 확보 및 검증을 위한 사실 확인, 신뢰도 점수 및 의미론적 유사성 검색 수행, 구조화된 출력을 적용하는 JSON Schema 사용)
- 기술 3.1.4: 심층 방어 안전 시스템을 구축하여 FM 오용에 대한 포괄적인 보호 제공(예: Amazon Comprehend를 사용하여 사전 처리 필터 개발, Amazon Bedrock을 사용하여 모델 기반 가드레일 구현, Lambda 함수를 사용하여 사후 처리 검증 수행, API 응답 필터링을 구현하기 위해 API Gateway 사용)
- 기술 3.1.5: 지능형 위협 탐지를 구현하여 적대적 입력 및 보안 취약성으로부터 보호(예: 프롬프트 인젝션 및 탈옥 탐지 메커니즘, 입력 삭제 및 콘텐츠 필터, 안전 분류기, 자동화된 적대적 테스트 워크플로 사용)

작업 3.2: 데이터 보안 및 개인 정보 보호 제어 구현

- 기술 3.2.1: 보호된 AI 환경을 개발하여 FM 배포를 위한 포괄적인 보안 보장(예: VPC 엔드포인트를 사용하여 네트워크 격리, IAM 정책을 사용하여 보안 데이터 액세스 패턴 적용, AWS Lake Formation을 사용하여 세분화된 데이터 액세스 제공, CloudWatch를 사용하여 데이터 액세스 모니터링)
- 기술 3.2.2: FM 상호 작용 중에 민감한 정보를 보호하기 위한 개인 정보 보호 시스템 개발(예: Amazon Comprehend 및 Amazon Macie를 사용하여 개인 식별 정보(PII) 탐지, Amazon Bedrock 네이티브 데이터 프라이버시 기능, 출력을 필터링하는 Amazon Bedrock 가드레일 사용, 데이터 보존 정책을 구현하기 위한 Amazon S3 수명 주기 구성)
- 기술 3.2.3: 프라이버시 중심의 AI 시스템을 구축하여 FM 유용성과 효율성을 유지하면서 사용자 개인 정보 보호(예: 데이터 마스킹 기술, Amazon Comprehend PII 탐지, 민감한 정보에 대한 익명화 전략, Amazon Bedrock 가드레일 사용)

작업 3.3: AI 거버넌스 및 규정 준수 메커니즘 구현

- 기술 3.3.1: FM 배포의 규정 준수를 보장하기 위한 규정 준수 프레임워크 개발(예: SageMaker AI를 사용하여 프로그래밍 모델 카드 개발, AWS Glue를 사용하여 데이터 계보 자동 추적, 체계적인 데이터 소스 속성을 위한 메타데이터 태깅, CloudWatch Logs를 사용하여 포괄적인 의사 결정 로그 수집)
- 기술 3.3.2: GenAI 애플리케이션에서 추적성을 유지하기 위해 데이터 소스 추적 구현(예: AWS Glue Data Catalog를 사용하여 데이터 소스 등록, FM 생성 콘텐츠의 소스 속성을 위한 메타데이터 태깅, 감사 로깅에 CloudTrail 사용)
- 기술 3.3.3: FM 구현을 일관되게 감독할 수 있는 조직 거버넌스 시스템 구축(예: 조직 정책, 규제 요구 사항 및 책임 있는 AI 원칙에 부합하는 포괄적인 프레임워크 사용)
- 기술 3.3.4: 지속적인 모니터링 및 고급 거버넌스 제어를 구현하여 안전 감사 및 규제 대비 지원(예: 오용, 드리프트 및 정책 위반에 대한 자동 감지, 바이어스 드리프트 모니터링, 자동화된 경고 및 수정 워크플로, 토큰 수준 수정, 응답 로깅, AI 출력 정책 필터 사용)

작업 3.4: 책임 있는 AI 원칙 구현

- 기술 3.4.1: FM 출력에서 투명한 AI 시스템 개발(예: 추론 디스플레이를 사용하여 사용자 대상 설명 제공, CloudWatch를 사용하여 신뢰도 지표 수집 및 불확실성 정량화, 소스 속성을 위한 증거 제시, 추론 추적을 제공하는 Amazon Bedrock 에이전트 추적 사용)
- 기술 3.4.2: 공정성 평가를 적용하여 편향되지 않은 FM 출력 보장(예: CloudWatch, Amazon Bedrock Prompt Management 및 Amazon Bedrock Prompt Flows에서 사전 정의된 공정성 지표를 사용하여 체계적인 A/B 테스트 수행, 평가형 LLM 솔루션을 갖춘 Amazon Bedrock으로 자동화된 모델 평가 수행)
- 기술 3.4.3: 책임 있는 AI 사례를 준수하기 위한 정책 준수 AI 시스템 개발(예: 정책 요구 사항에 기반한 Amazon Bedrock 가드레일, FM 제한을 문서화하는 모델 카드, 자동화된 규정 준수 검사를 수행하는 Lambda 함수 사용)

콘텐츠 도메인 4: GenAI 애플리케이션을 위한 운영 효율성 및 최적화

작업 4.1: 비용 최적화 및 리소스 효율성 전략 구현

- 기술 4.1.1: 효율성을 유지하면서 FM 비용을 줄일 수 있는 토큰 효율성 시스템 개발(예: 토큰 추정 및 추적, 컨텍스트 창 최적화, 응답 크기 제어, 프롬프트 압축, 컨텍스트 프루닝, 응답 제한 사용)

- 기술 4.1.2: 비용 효율적인 모델 선택 프레임워크 만들기(비용 대비 성능 절충 평가, 쿼리 복잡성에 따른 계층화된 FM 사용, 응답 품질에 대한 추론 비용 균형 조정, 가격 대비 성능 비율 측정, 효율적인 추론 패턴 사용)
- 기술 4.1.3: GenAI 워크로드의 리소스 활용률 및 처리량을 극대화하는 고성능 FM 시스템 개발(예: 배치 처리 전략, 용량 계획, 사용률 모니터링, 오토 스케일링 구성, 프로비저닝된 처리량 최적화 사용)
- 기술 4.1.4: 지능형 캐싱 시스템을 구축하여 불필요한 FM 호출을 피함으로써 비용을 줄이고 응답 시간 개선(예: 시맨틱 캐싱, 결과 핑거프린팅, 엣지 캐싱, 결정론적 요청 해싱, 프롬프트 캐싱 사용)

작업 4.2: 애플리케이션 성능 최적화

- 기술 4.2.1: 지연 시간 비용 절충을 해결하고 FM을 통한 사용자 경험을 개선하는 반응형 AI 시스템 만들기(예: 사전 계산을 사용하여 예측 가능한 쿼리 수행, 시간에 민감한 애플리케이션을 위한 지연 시간에 최적화된 Amazon Bedrock 모델, 복잡한 워크플로에 대한 병렬 요청, 응답 스트리밍, 성능 벤치마킹 사용)
- 기술 4.2.2: 검색 성능을 개선하여 FM 컨텍스트 증대를 위해 검색된 정보의 관련성 및 속도 개선(예: 인덱스 최적화, 쿼리 전처리, 사용자 지정 점수를 통한 하이브리드 검색 구현 사용)
- 기술 4.2.3: FM 처리량 최적화를 구현하여 GenAI 워크로드의 특정 처리량 문제 해결(예: 토큰 처리 최적화, 배치 추론 전략, 동시 모델 호출 관리 사용)
- 기술 4.2.4: FM 성능을 개선하여 특정 GenAI 사용 사례에 맞는 최적의 결과 확보(예: 모델별 파라미터 구성, 개선 사항 평가를 위한 A/B 테스트, 적절한 온도 및 요구 사항에 따른 top-k/top-p 선택 사용)
- 기술 4.2.5: FM 워크로드를 위한 효율적인 리소스 할당 시스템 만들기(예: 토큰 처리 요구 사항에 대한 용량 계획, 프롬프트 및 완료 패턴에 대한 사용률 모니터링, GenAI 트래픽 패턴에 최적화된 오토 스케일링 구성 사용)
- 기술 4.2.6: GenAI 워크플로의 FM 시스템 성능 최적화(예: 프롬프트 완료 패턴을 위한 API 호출 프로파일링, 검색 증대를 위한 벡터 데이터베이스 쿼리 최적화, LLM 추론에 특화된 지연 시간 감소 기술, 효율적인 서비스 통신 패턴 사용)

작업 4.3: GenAI 애플리케이션을 위한 모니터링 시스템 구현

- 기술 4.3.1: FM 애플리케이션 성능에 대한 완전한 가시성을 제공하는 총체적 관찰성 시스템 구축(예: 운영 지표, 성능 추적, FM 상호 작용 추적, 사용자 지정 대시보드가 포함된 비즈니스 영향 지표 사용
- 기술 4.3.2: 포괄적인 GenAI 모니터링 시스템을 구현하여 문제를 사전에 식별하고 FM 구현과 관련된 주요 성과 지표 평가(예: CloudWatch를 사용하여 토큰 사용, 프롬프트 효과, 할루시네이션

비율, 응답 품질, 토큰 버스트 패턴 및 응답 드리프트에 대한 이상 탐지, Amazon Bedrock Model Invocation Logs를 사용하여 상세한 요청 및 응답 분석, 성능 벤치마크, 비용 이상 탐지 수행)

- 기술 4.3.3: FM 애플리케이션에 실행 가능한 인사이트를 제공하는 통합 관찰성 솔루션 개발(예: 운영 지표 대시보드, 비즈니스 영향 시각화, 규정 준수 모니터링, 포렌식 추적성 및 감사 로깅, 사용자 상호작용 추적, 모델 행동 패턴 추적 사용)
- 기술 4.3.4: FM을 위한 최적의 도구 운영 및 활용을 보장하는 도구 성능 프레임워크 만들기(예: 호출 패턴 추적, 성능 지표 수집, 도구 호출 관찰성 및 다중 에이전트 조정 추적, 이상 탐지를 위한 사용 기준 사용)
- 기술 4.3.5: FM 증대를 위한 최적의 벡터 저장소 운영 및 신뢰성을 보장하는 벡터 저장소 운영 관리 시스템 구축(예: 벡터 데이터베이스 성능 모니터링, 자동화된 인덱스 최적화 루틴, 데이터 품질 검증 프로세스 사용)
- 기술 4.3.6: FM별 문제 해결 프레임워크를 개발하여 기존 ML 시스템에는 없는 고유한 GenAI 실패 모드 식별(예: 골든 데이터세트를 사용하여 할루시네이션 탐지, 비교 기술을 출력하여 응답 일관성 분석 수행, 추론 경로 추적을 통한 논리적 오류 식별, 특수 관찰성 파이프라인)

콘텐츠 도메인 5: 테스트, 검증 및 문제 해결

작업 5.1: GenAI용 평가 시스템 구현

- 기술 5.1.1: 포괄적인 평가 프레임워크를 개발하여 기존 ML 평가 접근 방식을 넘어서는 FM 출력의 품질 및 효과 평가(예: 관련성, 사실적 정확성, 일관성 및 유창성에 대한 지표 사용)
- 기술 5.1.2: 체계적인 모델 평가 시스템을 구축하여 최적의 구성 파악(예: Amazon Bedrock Model Evaluations, FM의 A/B 테스트 및 카나리 테스트, 다중 모델 평가, 토큰 효율성, 지연 시간 대비 품질 비율, 비즈니스 성과 측정을 위한 비용 성능 분석 사용)
- 기술 5.1.3: 사용자 경험을 기반으로 FM 성능을 지속적으로 개선하기 위한 사용자 중심 평가 메커니즘 개발(예: 피드백 인터페이스, 모델 출력용 등급 시스템, 응답 품질 평가를 위한 주석 워크플로 사용)
- 기술 5.1.4: FM에 대한 일관된 성능 표준을 유지하는 체계적인 품질 보증 프로세스 만들기(예: 지속적 평가 워크플로, 모델 출력에 대한 회귀 테스트, 배포를 위한 자동화된 품질 게이트 사용)
- 기술 5.1.5: FM 출력에 대해 다양한 관점에서 철저한 평가를 보장하는 포괄적인 평가 시스템 개발(예: RAG 평가, 평가형 LLM 기술을 사용한 자동 품질 평가, 사람의 피드백 수집 인터페이스 사용)
- 기술 5.1.6: 검색 품질 테스트를 구현하여 FM 증강을 위한 정보 검색 구성 요소 평가 및 최적화(예: 관련성 점수 매칭, 컨텍스트 매칭 검증, 검색 지연 시간 측정 사용)

- 기술 5.1.7: 에이전트가 정확하고 효율적으로 작업을 수행할 수 있도록 에이전트 성능 프레임워크 개발(예: 작업 완료율 측정, 도구 사용 효율성 평가, Amazon Bedrock Agent 평가, 다단계 워크플로의 추론 품질 평가 사용)
- 기술 5.1.8: 포괄적인 보고 시스템을 만들어 FM 구현을 위해 이해 관계자에게 성과 지표와 인사이트를 효과적으로 전달(예: 시각화 도구, 자동화된 보고 메커니즘, 모델 비교 시각화 사용)
- 기술 5.1.9: FM 업데이트 중에 신뢰성을 유지하기 위한 배포 검증 시스템 만들기(예: 합성 사용자 워크플로, 할루시네이션 비율 및 시맨틱 드리프트에 대한 AI 관련 출력 검증, 응답 일관성 보장을 위한 자동화된 품질 검사 사용)

작업 5.2: GenAI 애플리케이션 문제 해결

- 기술 5.2.1: 콘텐츠 처리 문제를 해결하여 필요한 정보가 FM 상호 작용에서 완전히 처리되도록 설정(예: 컨텍스트 창 오버플로 진단, 동적 청킹 전략, 프롬프트 설계 최적화, 잘림 관련 오류 분석 사용)
- 기술 5.2.2: FM 통합 문제를 진단하고 해결하여 GenAI 서비스와 관련된 API 통합 문제 식별 및 해결(예: 오류 로깅, 요청 검증, 응답 분석 사용)
- 기술 5.2.3: 프롬프트 엔지니어링 문제를 해결하여 기본적인 프롬프트 조정 이상의 FM 응답 품질 및 일관성 개선(예: 프롬프트 테스트 프레임워크, 버전 비교, 체계적 개선 사용)
- 기술 5.2.4: 검색 시스템 문제를 해결하여 FM 증강을 위한 정보 검색 효과에 영향을 미치는 문제 식별 및 해결(예: 모델 응답 관련성 분석, 임베딩 품질 진단, 드리프트 모니터링, 벡터화 문제 해결, 청킹 및 전처리 개선, 벡터 검색 성능 최적화 사용)
- 기술 5.2.5: 즉각적인 유지 관리 문제를 해결하여 FM 상호 작용의 성능 지속적으로 개선(예: 템플릿 테스트 및 CloudWatch Logs를 사용하여 프롬프트 혼동 문제 진단, X-Ray를 사용하여 즉각적인 관찰성 파이프라인 구현, 형식 불일치를 탐지하기 위한 스키마 검증, 체계적인 프롬프트 개선 워크플로 사용)

시험에 출제될 수 있는 기술 및 개념

다음 목록에는 시험에 출제될 수 있는 기술 및 개념이 포함되어 있습니다. 이 목록에 모든 사항이 포함된 것은 아니며 변경될 수 있습니다. 이 목록에 나와 있는 다음 항목의 배치와 순서가 시험에서의 상대적 가중치 또는 중요도를 의미하지는 않습니다.

- 검색 증강 생성(RAG)
- 벡터 데이터베이스 및 임베딩
- 프롬프트 엔지니어링 및 관리

- 파운데이션 모델(FM) 통합
- 에이전틱 AI 시스템
- 책임 있는 AI 사례
- 콘텐츠 안전 및 조정
- 모델 평가 및 검증
- AI 워크로드의 비용 최적화
- AI 애플리케이션을 위한 성능 튜닝
- AI 시스템의 모니터링 및 관찰성
- AI 애플리케이션을 위한 보안 및 거버넌스
- API 설계 및 통합 패턴
- 이벤트 기반 아키텍처
- 서버리스 컴퓨팅
- 컨테이너 오케스트레이션
- 코드형 인프라(IaC)
- AI 애플리케이션을 위한 CI/CD
- 하이브리드 클라우드 아키텍처
- 엔터프라이즈 시스템 통합

시험에서 언급되는 AWS 서비스

AWS Certification에서는 약어나 괄호 정보가 포함되어 있는 잘 알려진 AWS 서비스 이름에 대해 공식 약칭을 사용하여 이 시험에서 읽기에 대한 부담을 덜어줍니다. 예를 들어 Amazon Simple Notification Service(Amazon SNS)는 시험에서 Amazon SNS로 표시됩니다.

시험의 도움말 기능(모든 문항에 사용 가능)에는 약칭으로 된 AWS 서비스 이름과 해당하는 전체 이름 목록이 포함되어 있습니다.

시험에서 약칭으로 표시되는 AWS 서비스 목록은 AWS Certification 웹 사이트의 서비스 이름을 참조하십시오. 목록에 있지만 시험 범위에 포함되지 않는 서비스는 시험에 표시되지 않습니다.

> **Note**
>
> 시험에서 일부 약어는 완전히 풀어 쓰여 있지 않거나 도움말 기능에서 제공되지 않을 수 있습니다. 일부 AWS 서비스의 공식 전체 이름에는 풀어 쓰지 않는 약어가 포함되어 있습니다(예:

Amazon API Gateway, Amazon EMR). 또한 시험에는 대상 응시자가 알아야 할 다른 약어가 포함될 수 있습니다.

범위 내 AWS 서비스

다음 목록은 시험 범위에 포함되는 AWS 서비스 및 기능입니다. 이 목록은 전체 목록이 아니며 변경될 수 있습니다. AWS 제품 및 서비스는 해당 제품 및 서비스의 주요 기능에 맞는 카테고리로 표시됩니다.

주제

- 분석
- 애플리케이션 통합
- 컴퓨팅
- 컨테이너
- 고객 지원
- 데이터베이스
- 개발자 도구
- 기계 학습
- AWS의 관리 및 거버넌스
- 마이그레이션 및 전송
- 네트워킹 및 콘텐츠 전송
- 보안, ID 및 규정 준수
- 스토리지

분석

- Amazon Athena
- Amazon EMR
- AWS Glue
- Amazon Kinesis
- Amazon OpenSearch Service
- Amazon Quick Sight
- Amazon Managed Streaming for Apache Kafka(Amazon MSK)

애플리케이션 통합

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

컴퓨팅

- AWS App Runner
- Amazon EC2
- AWS Lambda
- AWS Lambda@Edge
- AWS Outposts
- AWS Wavelength

컨테이너

- Amazon ECR
- Amazon ECS
- Amazon EKS
- AWS Fargate

고객 지원

- Amazon Connect

데이터베이스

- Amazon Aurora

- Amazon DocumentDB
- Amazon DynamoDB
- Amazon DynamoDB Streams
- Amazon ElastiCache
- Amazon Neptune
- Amazon RDS

개발자 도구

- AWS Amplify
- AWS CDK
- AWS CLI
- AWS CloudFormation
- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline
- Kiro
- AWS 도구 및 SDK
- AWS X-Ray

기계 학습

- Amazon Augmented AI
- Amazon Bedrock
- Amazon Bedrock AgentCore
- Amazon Bedrock Knowledge Bases
- Amazon Bedrock Prompt Management
- Amazon Bedrock Prompt Flows
- Amazon Comprehend
- Amazon Kendra

- Amazon Lex
- Amazon Q Business
- Amazon Q Business Apps
- Amazon Q Developer
- Amazon Quick
- Amazon Rekognition
- Amazon SageMaker AI
- Amazon SageMaker Clarify
- Amazon SageMaker Data Wrangler
- Amazon SageMaker Ground Truth
- Amazon SageMaker JumpStart
- Amazon SageMaker Model Monitor
- Amazon SageMaker Model Registry
- Amazon SageMaker Neo
- Amazon SageMaker Processing
- Amazon SageMaker Unified Studio
- Amazon Textract
- Amazon Titan
- Amazon Transcribe

AWS의 관리 및 거버넌스

- AWS Auto Scaling
- AWS Chatbot
- AWS CloudTrail
- Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon CloudWatch Synthetics
- AWS Cost Anomaly Detection
- AWS Cost Explorer

- Amazon Managed Grafana
- AWS Service Catalog
- AWS Systems Manager
- AWS Well-Architected Tool

마이그레이션 및 전송

- AWS DataSync
- AWS Transfer Family

네트워킹 및 콘텐츠 전송

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

보안, ID 및 규정 준수

- Amazon Cognito
- AWS Encryption SDK
- IAM
- IAM Access Analyzer
- IAM Identity Center
- AWS KMS
- Amazon Macie
- AWS Secrets Manager

- AWS WAF

스토리지

- Amazon EBS
- Amazon EFS
- Amazon S3
- Amazon S3 Intelligent-Tiering
- Amazon S3 수명 주기 정책
- Amazon S3 크로스 리전 복제

범위 외 AWS 서비스

다음 목록은 시험 범위에 포함되지 않는 AWS 서비스 및 기능입니다. 이 목록은 전체 목록이 아니며 변경될 수 있습니다. 시험의 대상 작업 역할과 전혀 관련이 없는 AWS 제품 및 서비스는 다음 목록에서 제외됩니다.

주제

- 애플리케이션 통합
- 분석
- 블록체인
- 비즈니스 애플리케이션
- 클라우드 재무 관리
- 컴퓨팅
- 컨테이너
- 고객 지원
- 데이터베이스
- 개발자 도구
- 최종 사용자 컴퓨팅
- 프런트엔드 웹 및 모바일
- 게임 개발

- 사물 인터넷(IoT)
- AWS 관리 및 거버넌스
- 기계 학습
- 미디어 서비스
- 마이그레이션 및 전송
- 네트워킹 및 콘텐츠 전송
- 양자 기술
- 로보틱스
- 인공위성

애플리케이션 통합

- Amazon MQ

분석

- AWS Clean Rooms
- AWS Data Exchange
- Amazon DataZone
- Amazon FinSpace

블록체인

- Amazon Managed Blockchain(AMB)

비즈니스 애플리케이션

- Alexa for Business
- Amazon Chime
- AWS Wickr
- Amazon WorkDocs
- Amazon WorkMail

클라우드 재무 관리

- AWS Budgets
- AWS Cost and Usage Report
- 예약 인스턴스 보고
- AWS 절감형 플랜

컴퓨팅

- AWS Batch
- Amazon EC2 Image Builder
- Amazon ECS Anywhere
- Amazon EKS Anywhere
- AWS Elastic Beanstalk
- Amazon Lightsail
- AWS Local Zones
- AWS Serverless Application Repository

컨테이너

- AWS App2Container
- AWS Copilot
- Red Hat OpenShift Service on AWS(ROSA)

고객 지원

- Amazon SES

데이터베이스

- Amazon Keyspaces
- Amazon Quantum Ledger Database(QLDB)

- Amazon Redshift
- Amazon Timestream

개발자 도구

- AWS Cloud9
- AWS CloudShell
- Amazon CodeGuru
- AWS CodeStar
- Amazon Corretto

최종 사용자 컴퓨팅

- Amazon AppStream 2.0
- Amazon WorkLink
- Amazon WorkSpaces
- Amazon WorkSpaces Web

프런트엔드 웹 및 모바일

- AWS Device Farm
- Amazon Location Service
- Amazon Pinpoint

게임 개발

- Amazon GameLift
- Amazon Lumberyard

사물 인터넷(IoT)

- AWS IoT 1-Click

- AWS IoT Analytics
- AWS IoT Button
- AWS IoT Core
- AWS IoT Device Defender
- AWS IoT Device Management
- AWS IoT Events
- AWS IoT FleetWise
- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS IoT TwinMaker

AWS 관리 및 거버넌스

- AWS Console Mobile Application
- AWS Health Dashboard
- AWS License Manager
- AWS Proton
- AWS Trusted Advisor

기계 학습

- AWS DeepComposer
- AWS DeepRacer
- Amazon DevOps Guru
- Amazon Forecast
- Amazon Fraud Detector
- Amazon HealthLake
- Amazon Lookout for Equipment
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon Monitron

- AWS Panorama

미디어 서비스

- Amazon Elastic Transcoder
- AWS Elemental MediaConnect
- AWS Elemental MediaConvert
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaStore
- AWS Elemental MediaTailor
- Amazon Interactive Video Service
- Amazon Kinesis Video Streams
- Amazon Nimble Studio

마이그레이션 및 전송

- AWS Application Discovery Service
- AWS Application Migration Service
- CloudEndure Migration
- AWS Migration Hub
- AWS Snow Family

네트워킹 및 콘텐츠 전송

- AWS App Mesh
- AWS Cloud Map
- AWS Direct Connect
- AWS Private 5G
- AWS Transit Gateway
- AWS VPN

양자 기술

- Amazon Braket

로보틱스

- AWS RoboMaker

인공위성

- AWS Ground Station

설문 조사

이 시험 안내서가 도움이 되었습니까? 설문 조사에 참여하여 알려 주시기 바랍니다.