aws training and certification

AWS Certified Developer – Associate (DVA-C01) Prüfungsleitfaden

Einführung

Die Prüfung AWS Certified Developer – Associate (DVA-C01) ist für Personen gedacht, die eine Entwicklerrolle ausüben. Die Prüfung bewertet die Fähigkeiten des Kandidaten in folgenden Bereichen:

- Kenntnisse über wichtige AWS-Services, Anwendungsfälle sowie bewährte Methoden hinsichtlich der grundlegenden AWS-Architektur
- Kenntnisse in der Entwicklung, Bereitstellung und Fehlerbehebung cloudbasierter Anwendungen mit AWS

Beschreibung der Zielkandidaten

Zielkandidaten sollten über mindestens 1 Jahr praktische Erfahrung in der Entwicklung und Wartung einer AWS-basierten Anwendung verfügen.

Empfohlene allgemeine IT-Kenntnisse

Die Zielkandidaten sollte über Folgendes verfügen:

- Fundierte Kenntnisse mindestens einer höheren Programmiersprache
- Verständnis des Anwendungslebenszyklus-Managements
- Die Fähigkeit, Code für serverlose Anwendungen zu schreiben
- Verständnis der Verwendung von Containern im Entwicklungsprozess

Empfohlenes AWS-Wissen

Die Zielkandidaten sollten Folgendes können:

- Mithilfe von AWS-Service-APIs, CLI und Software Development Kits (SDKs) Anwendungen schreiben
- Wichtige Funktionen der AWS-Services identifizieren
- Das AWS-Modell der geteilten Verantwortung verstehen
- Die Pipeline für Continuous Integration und Continuous Delivery (CI/CD) verwenden, um Anwendungen auf AWS bereitzustellen
- AWS-Services verwenden und damit interagieren
- Auf Basis grundlegender Kenntnisse von cloudnativen Anwendungen Code schreiben
- Code unter Verwendung bewährter AWS-Sicherheitsmethoden schreiben (indem beispielsweise anstelle von geheimen und Zugriffsschlüsseln IAM-Rollen im Code verwendet werden)
- Codemodule in AWS erstellen, warten und debuggen

Was ist für den Zielkandidaten nicht relevant?

Es folgt eine nicht vollständige Liste von zugehörigen Aufgaben, deren Durchführung vom Zielkandidaten nicht erwartet wird. Diese Punkte kommen in der Prüfung nicht vor:

- Entwicklung von Architekturen (z. B. verteilte Systeme, Microservices)
- Entwicklung und Implementierung von CI/CD-Pipelines
- Verwaltung von IAM-Benutzern und -Gruppen
- Verwaltung des Amazon Elastic Container Service (Amazon ECS)
- Entwicklung der AWS-Netzwerkinfrastruktur (z. B. Amazon VPC, AWS Direct Connect)
- Kenntnisse über Compliance und Lizenzierung

Eine detaillierte Liste spezifischer Tools und Technologien, die möglicherweise in der Prüfung behandelt werden, sowie Listen der relevanten und nicht relevanten AWS-Services finden Sie im Anhang.

Inhalt der Prüfung

Antwortarten

Es gibt zwei Arten von Fragen bei der Prüfung:

- **Multiple Choice:** beinhaltet eine richtige und drei falsche Antworten (Distraktoren)
- **Mehrfache Antwort:** beinhaltet zwei oder mehr korrekte Antworten aus fünf oder mehr Antwortoptionen

Wählen Sie eine oder mehrere Antworten aus, die die Aussage am besten vervollständigen oder die Frage beantworten. Distraktoren oder falsche Antworten sind Antwortoptionen, für die sich ein Kandidat mit unvollständigem Wissen oder fehlenden Fähigkeiten entscheiden könnte. Distraktoren sind im Allgemeinen plausible Antworten, die zum Inhaltsbereich passen.

Unbeantwortete Fragen werden als falsch gewertet. Raten ist ohne Abzüge möglich. Die Prüfung beinhaltet 50 Fragen, die sich auf Ihren Punktestand auswirken.

Nicht gewerteter Inhalt

Die Prüfung umfasst 15 nicht gewertete Fragen, die sich nicht auf Ihre Punktzahl auswirken. AWS sammelt mit diesen nicht gewerteten Fragen Informationen zur Kandidatenleistung, um sie in Zukunft möglicherweise als gewertete Fragen zu verwenden. Diese nicht gewerteten Fragen werden in der Prüfung nicht identifiziert.

Prüfungsergebnisse

Die Prüfung AWS Certified Developer – Associate (DVA-C01) ist eine sogenannte Pass/Fail-Prüfung. Die Prüfung wird nach einem Mindeststandard bewertet, der von AWS-Experten festgelegt wurde, die die bewährten Methoden und Richtlinien der Zertifizierungsbranche befolgen.

Ihre Prüfungsergebnisse werden als skalierter Punktwert zwischen 100 und 1.000 angezeigt. Die erforderliche Mindestpunktzahl beträgt 720. Ihre Punktzahl zeigt, wie Sie die Prüfung insgesamt absolviert haben und ob Sie bestanden haben oder nicht. Skalierte Bewertungsmodelle helfen dabei, Ergebnisse mehrerer Prüfungsformen mit möglicherweise leicht unterschiedlichen Schwierigkeitsgraden gleichzusetzen.

Ihr Bewertungsbericht enthält eine Tabelle mit Klassifizierungen Ihrer Leistung auf jeder Abschnittsebene. Diese Informationen sollen allgemeines Feedback zu Ihrer Prüfungsleistung geben. Für die Prüfung wird ein kompensatorisches Bewertungsmodell verwendet, was bedeutet, dass Sie nicht in jedem Abschnitt die erforderliche Punktzahl erreichen müssen. Sie müssen nur die Gesamtprüfung bestehen.

Jeder Abschnitt der Prüfung hat eine bestimmte Gewichtung. Daher enthalten einige Abschnitte mehr Fragen als andere. Die Tabelle enthält allgemeine Informationen, die Ihre Stärken und Schwächen hervorheben. Bewerten Sie das Feedback auf Abschnittsebene mit Bedacht.

Inhaltsgliederung

Dieser Prüfungsleitfaden enthält Gewichtungen, Testbereiche und Ziele für die Prüfung. Es handelt sich nicht um eine umfassende Auflistung der Prüfungsinhalte. Für jedes der Ziele steht jedoch zusätzlicher Kontext zur Verfügung, der Ihnen bei der Vorbereitung auf die Prüfung hilft. In der folgenden Tabelle sind die wichtigsten Inhaltsbereiche und ihre Gewichtungen aufgeführt. Auf die Tabelle folgt eine vollständige Übersicht über den Prüfungsinhalt, die den zusätzlichen Kontext enthält. Der Prozentsatz in jedem Bereich spiegelt nur die gewerteten Inhalte wider.

Bereich	% der Prüfung
Bereich 1: Bereitstellung	22 %
Bereich 2: Sicherheit	26 %
Bereich 3: Entwicklung mithilfe der AWS-Services	30 %
Bereich 4: Refactoring	10 %
Bereich 5: Überwachung und Fehlerbehebung	12 %
GESAMT	**100 %**

Bereich 1: Bereitstellung

1.1 Geschriebenen Code anhand vorhandener CI/CD-Pipelines, Prozesse und Muster in AWS bereitstellen.

- Code in ein Repository übertragen und Build-, Test- und/oder Bereitstellungsaktionen aufrufen
- Kennzeichnungen und Zweige für die Versionsverwaltung verwenden
- AWS CodePipeline verwenden, um Workflows in verschiedenen Umgebungen zu steuern
- AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeStar und AWS CodeDeploy für CI/CD-Zwecke anwenden
- Rollback-Plan ausführen, der auf den Richtlinien zur Anwendungsbereitstellung beruht

1.2 Anwendungen mithilfe von AWS Elastic Beanstalk bereitstellen.

- Vorhandene unterstützte Umgebungen verwenden, um einen neuen Anwendungsstapel zu definieren
- Anwendung verpacken
- Eine neue Anwendungsversion in die Elastic Beanstalk-Umgebung einführen
- Eine Bereitstellungsrichtlinie verwenden, um eine Anwendungsversion bereitzustellen (d. h. alles gleichzeitig, regelmäßig, regelmäßig mit Batch, unveränderlich)
- Den Anwendungszustand mithilfe des Elastic Beanstalk-Dashboards validieren
- Amazon CloudWatch Logs verwenden, um die Anwendungsprotokollierung zu konfigurieren

1.3 Das Anwendungsbereitstellungspaket vorbereiten, das in AWS bereitgestellt werden soll.
- Abhängigkeiten des Code-Moduls innerhalb des Pakets verwalten (wie Umgebungsvariablen, Konfigurationsdateien und statische Bilddateien)
- Paket-/Container-Verzeichnisstruktur aufzeigen und die Dateien entsprechend organisieren
- Anforderungen an die Anwendungsressourcen in AWS-Infrastrukturparameter umsetzen (z. B. Speicher, Kerne)

1.4 Serverlose Anwendungen bereitstellen.
- Eine Vorlage für ein AWS Serverless Application Model (AWS SAM) für einen vorgegebenen Anwendungsfall implementieren und starten
- Umgebungen in einzelnen AWS-Services verwalten (z. B. zwischen Entwicklung, Test und Produktion in Amazon API Gateway unterscheiden)

Bereich 2: Sicherheit

2.1 Authentifizierte Anrufe an AWS-Services tätigen.
- Erforderliche Richtlinien basierend auf den geringsten von der Anwendung erforderlichen Berechtigungen kommunizieren
- Eine IAM-Rolle annehmen, um auf einen Dienst zuzugreifen
- Den Anmeldeinformationsanbieter des Software Development Kits (SDK) vor Ort oder in der Cloud verwenden, um auf AWS-Services zuzugreifen (lokale Anmeldeinformationen im Vergleich zu Instance-Rollen)

2.2 Verschlüsselung mithilfe von AWS-Services implementieren.
- Daten im Ruhezustand mit AWS-Services verschlüsseln (clientseitig; serverseitig; Umschlagverschlüsselung)
- Daten während der Übertragung verschlüsseln

2.3 Anwendungsauthentifizierung und -autorisierung implementieren.
- Benutzeranmelde- und Registrierungsfunktionen für Anwendungen mit Amazon Cognito-Identitäts- oder Benutzerpools hinzufügen
- Mithilfe von Anmeldeinformationen, die von Amazon Cognito bereitgestellt wurden, Code schreiben, um auf AWS-Services zuzugreifen
- Amazon Cognito-Synchronisierung verwenden, um Benutzerprofile und Daten zu synchronisieren
- Von Entwicklern authentifizierte Identitäten verwenden, um zwischen Endbenutzergeräten, Backend-Authentifizierung und Amazon Cognito zu interagieren

Bereich 3: Entwicklung mithilfe der AWS-Services

3.1 Code für serverlose Anwendungen schreiben.
- Serverbasierte und serverlose Modelle vergleichen und gegenüberstellen (z. B. Microservices, die zustandslose Beschaffenheit serverloser Anwendungen, Skalierung serverloser Anwendungen und Entkopplung von Ebenen in serverlosen Anwendungen)
- AWS Lambda-Funktionen durch Definieren der Umgebungsvariablen und -parameter konfigurieren (z. B. Speicher, Timeout, Laufzeit, Handler)
- Einen API-Endpunkt mit Amazon API Gateway erstellen
- Geeignete API-Aktionen wie GET, POST mit dem API-Endpunkt erstellen und testen
- Amazon DynamoDB-Konzepte anwenden (z. B. Tabellen, Elemente und Attribute)
- Lese-/Schreibkapazitätseinheiten für Amazon DynamoDB basierend auf Anwendungsanforderungen berechnen

- Eine AWS Lambda-Funktion mit einer AWS-Ereignisquelle verknüpfen (z. B. Amazon API Gateway, Amazon CloudWatch Event, Amazon S3 Events, Amazon Kinesis)
- Eine AWS Lambda-Funktion synchron und asynchron aufrufen

3.2 Funktionale Anforderungen in Anwendungsdesign umsetzen.

- Für einen bestimmten Anwendungsfall Echtzeitverarbeitung vs. Batchverarbeitung festlegen
- Für einen bestimmten Anwendungsfall die synchrone vs. asynchrone Verwendung festlegen
- Für einen bestimmten Anwendungsfall die Verwendung des Ereignisses vs. des Zeitplans/der Abfrage festlegen
- Kompromisse für Konsistenzmodelle in einem Anwendungsdesign berücksichtigen

3.3 Das Anwendungsdesign in den Anwendungscode implementieren.

- Code schreiben, um Messaging-Services (z. B. SQS, SNS) zu nutzen
- Amazon ElastiCache verwenden, um einen Datenbank-Cache zu erstellen
- Amazon DynamoDB verwenden, um Objekte in Amazon S3 zu indizieren
- Eine zustandslose AWS Lambda-Funktion schreiben
- Eine Webanwendung mit zustandslosen Webservern schreiben (Status externalisieren)

3.4 Mithilfe von APIs, SDKs und AWS CLI Code schreiben, der mit AWS-Services interagiert.

- Geeignete APIs, Software Development Kits (SDKs) und CLI-Befehle für die Codekomponenten auswählen
- Belastbaren Code schreiben, der mit Fehlern oder Ausnahmen umgehen kann (d. h. Wiederholungen mit exponentiellem Backoff und Jitter)

Bereich 4: Refactoring

4.1 Anwendungen optimieren, um AWS-Services und Funktionen optimal zu nutzen.

- AWS Caching-Services implementieren, um die Leistung zu optimieren (z. B. Amazon ElastiCache, Amazon API Gateway-Cache)
- Ein Amazon S3-Benennungsschema für optimale Leseleistung anwenden

4.2 Vorhandenen Anwendungscode zur Ausführung auf AWS migrieren.

- Abhängigkeiten isolieren
- Anwendung als einen oder mehrere zustandslose Prozesse ausführen
- Entwicklungen so ausführen, dass sie horizontale Skalierbarkeit ermöglichen
- Status externalisieren

Bereich 5: Überwachung und Fehlerbehebung

5.1 Code schreiben, der sich überwachen lässt.

- Benutzerdefinierte Amazon CloudWatch-Metriken erstellen
- Protokollierung auf eine für Systembetreiber verfügbare Weise ausführen
- Quellcode der Anwendung konfigurieren, um die Nachverfolgung in AWS X-Ray zu aktivieren

5.2 Ursachenanalyse bei Fehlern durchführen, die beim Testen oder in der Produktion festgestellt wurden.

- Ausgabewerte des Protokollierungsmechanismus in AWS interpretieren, um Fehler in den Protokollen zu identifizieren
- Build- und Testverlauf in AWS-Services (z. B. AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline) überprüfen, um Probleme zu identifizieren
- AWS-Services (z. B. Amazon CloudWatch, VPC Flow Logs und AWS X-Ray) nutzen, um eine bestimmte fehlerhafte Komponente zu finden

Anhang

Welche Schlüsselinstrumente, Technologien und Konzepte könnten in der Prüfung behandelt werden?

Die folgende Liste enthält einige Tools und Technologien, die in der Prüfung abgefragt werden könnten. Diese Liste ist nicht vollständig und kann sich ändern. Sie soll Ihnen helfen, den allgemeinen Umfang der Services, Funktionen oder Technologien, die in der Prüfung vorkommen, zu verstehen. Die allgemeinen Tools und Technologien in dieser Liste sind in keiner bestimmten Reihenfolge aufgeführt. AWS-Services sind nach ihren primären Funktionen gruppiert. Auch wenn einige dieser Technologien wahrscheinlich ausführlicher in der Prüfung abgefragt werden als andere, sind deren Reihenfolge und Platzierung in dieser Liste kein Hinweis auf die relative Gewichtung oder Bedeutung:

- Analytik
- Anwendungsintegration
- Container
- Kosten- und Kapazitätsmanagement
- Datenbewegungen
- Entwickler-Tools
- Instances (virtuelle Maschinen)
- Management und Governance
- Netzwerk und Bereitstellung von Inhalten
- Sicherheit
- Serverless

AWS-Services und -Funktionen

Analytik:

- Amazon Elasticsearch Service (Amazon ES)
- Amazon Kinesis

Anwendungsintegration:

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

Computing:

- Amazon EC2
- AWS Elastic Beanstalk
- AWS Lambda

Container:

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

Datenbank:

- Amazon DynamoDB
- Amazon ElastiCache
- Amazon RDS

Entwickler-Tools:

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

Management und Governance:

- AWS CloudFormation
- Amazon CloudWatch

Netzwerk und Bereitstellung von Inhalten:

- Amazon API Gateway
- Amazon CloudFront
- Elastic Load Balancing

Sicherheit, Identität und Compliance:

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

Speicher:

- Amazon S3

AWS-Services und -Funktionen, die nicht in der Prüfung behandelt werden

Es folgt eine nicht vollständige Liste der AWS-Services und -Funktionen, die in der Prüfung nicht abgefragt werden. Diese Services und Funktionen stellen nicht alle AWS-Angebote dar, die vom Prüfungsinhalt ausgeschlossen sind. Services oder Funktionen, die in keinem Zusammenhang mit der durch die Prüfung angestrebten Position stehen, sind von der Liste ausgeschlossen, da sie als irrelevant angesehen werden.

Die folgenden AWS-Services und -Funktionen werden in der Prüfung nicht behandelt:

- 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-Familie
- AWS Storage Gateway
- AWS WAF
- Amazon WorkMail
- Amazon WorkSpaces