AW232 DevOps Engineering on AWS

Kurzbeschreibung:

Softwareentwickler, DevOps-Ingenieure und Cloud-Architekten lernen, wie sie mit DevOps-Prinzipien Anwendungen auf AWS effizient entwickeln, bereitstellen und betreiben. Vermittelt werden CI/CD, Infrastructure as Code, Microservices, Monitoring und Logging. Behandelt werden Multi-Pipeline-Workflows, Automatisierung mit CloudFormation sowie praxisnahe Übungen für serverlose und containerbasierte Anwendungen.

Zielgruppe:

Dieser Kurs **AW232 DevOps Engineering on AWS** richtet sich an:

- DevOps Engineers
- DevOps-Architekten
- Betriebsingenieure
- Systemadministratoren
- Entwickler

Voraussetzungen:

Um an dem Kurs **AW232 DevOps Engineering on AWS** bei qSkills teilnehmen zu können, sollten Sie die folgenden AWS-Trainings besucht haben:

- Systems Operations on AWS
- AW230 Developing on AWS

Darüber hinaus sollten Sie folgende Voraussetzungen erfüllen:

- Über Kenntnisse in einer oder mehreren Hochsprachen wie C#, Java, PHP, Ruby, Python verfügen
- Über mittlere Kenntnisse in der Administration von Linux- oder Windows-Systemen auf Kommandozeilenebene verfügen
- Zwei oder mehr Jahre Erfahrung in der Bereitstellung, dem Betrieb und der Verwaltung von AWS-Umgebungen besitzen

Sonstiges:

Dauer: 3 Tage

Preis: 1995 Euro plus Mwst.

Ziele:

In diesem Kurs **AW232 DevOps Engineering on AWS** werden Sie lernen:

- DevOps-Best Practices anzuwenden, um Anwendungen und Services mit hoher Geschwindigkeit auf AWS zu entwickeln, bereitzustellen und zu betreiben
- Vorteile, Rollen und Verantwortlichkeiten kleiner autonomer DevOps-Teams aufzulisten

- Eine Infrastruktur auf AWS zu entwerfen und zu implementieren, die DevOps-Entwicklungsprojekte unterstützt
- AWS Cloud9 zum Schreiben, Ausführen und Debuggen von Code zu nutzen
- Verschiedene Umgebungen mit AWS CloudFormation bereitzustellen
- Sichere, hochskalierbare und private Git-Repositories mit AWS CodeCommit zu hosten
- Git-Repositories in CI/CD-Pipelines zu integrieren
- Code-Build, -Test und -Packaging mit AWS CodeBuild zu automatisieren
- Docker-Images sicher zu speichern und in CI/CD-Pipelines zu integrieren
- CI/CD-Pipelines zum Bereitstellen von Anwendungen auf Amazon EC2, serverlosen Anwendungen und containerbasierten Anwendungen zu erstellen
- Gängige Bereitstellungsstrategien wie „All at Once“, „Rolling“ und „Blue/Green“ zu implementieren
- Tests und Sicherheitsmechanismen in CI/CD-Pipelines zu integrieren
- Anwendungen und Umgebungen mit AWS-Tools und -Technologien zu überwachen

Inhalte/Agenda:

- ♦ **Kursüberblick**
  - ♦ ◊ Kursziele
    ◊ Empfohlene Voraussetzungen
    ◊ Überblick über die Kursstruktur
- ♦ ◊
- ♦ **Einführung in DevOps**
  - ♦ ◊ Was ist DevOps?
    ◊ Die Amazon-Reise zu DevOps
    ◊ Grundlagen für DevOps
- ♦ ◊
- ♦ **Infrastrukturautomatisierung**
  - ♦ ◊ Einführung in Infrastrukturautomatisierung
    ◊ Einführung in das AWS CloudFormation-Template
    ◊ Bearbeitung eines AWS CloudFormation-Templates
    ◊ Demonstration: Struktur, Parameter, Stacks, Updates, Ressourcenimport und Drift-Erkennung in AWS CloudFormation
- ♦ ◊
- ♦ **AWS Toolkits**
  - ♦ ◊ Konfiguration der AWS CLI
    ◊ AWS Software Development Kits (AWS SDKs)
    ◊ AWS SAM CLI
    ◊ AWS Cloud Development Kit (AWS CDK)
    ◊ AWS Cloud9
    ◊ Demonstration: AWS CLI und AWS CDK
    ◊ Hands-on Lab: Bereitstellung und Verwaltung einer Basisinfrastruktur mit AWS CloudFormation
- ♦ ◊
- ♦ **Continuous Integration und Continuous Delivery (CI/CD) mit Entwickler-Tools**
  - ♦ ◊ CI/CD-Pipeline und Dev Tools
    ◊ Demonstration: CI/CD-Pipeline mit AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy und AWS CodePipeline
    ◊ Hands-on Lab: Bereitstellung einer Anwendung auf einer EC2-Flotte mit AWS CodeDeploy
    ◊ AWS CodePipeline
    ◊ Demonstration: AWS-Integration mit Jenkins
    ◊ Hands-on Lab: Automatisierte Code-Bereitstellung mit AWS CodePipeline
- ♦ ◊
- ♦ **Einführung in Microservices**
- ♦
- ♦ **DevOps und Container**
  - ♦ ◊ Bereitstellung von Anwendungen mit Docker
    ◊ Amazon Elastic Container Service und AWS Fargate
    ◊ Amazon Elastic Container Registry und Amazon Elastic Kubernetes Service
    ◊ Demonstration: CI/CD-Pipeline-Bereitstellung in einer containerisierten Anwendung
- ♦ ◊
- ♦ **DevOps und Serverless Computing**
  - ♦ ◊ AWS Lambda und AWS Fargate
    ◊ AWS Serverless Application Repository und AWS SAM
    ◊ AWS Step Functions
    ◊ Demonstration: Eigenschaften von AWS Lambda
    ◊ Demonstration: AWS SAM Quick Start in AWS Cloud9
    ◊ Hands-on Lab: Bereitstellung einer serverlosen Anwendung mit AWS SAM und einer CI/CD-Pipeline
- ♦ ◊
- ♦ **Bereitstellungsstrategien**
  - ♦ ◊ Continuous Deployment
    ◊ Bereitstellungen mit AWS-Services
- ♦ ◊
- ♦ **Automatisiertes Testen**
  - ♦ ◊ Einführung in das Testen
    ◊ Tests: Unit, Integration, Fault Tolerance, Load, Synthetic
    ◊ Produkt- und Serviceintegrationen
- ♦ ◊
- ♦ **Sicherheitsautomatisierung**
  - ♦ ◊ Einführung in DevSecOps
    ◊

◊ Sicherheit der Pipeline
◊ Sicherheit in der Pipeline
◊ Tools zur Bedrohungserkennung
◊ Demonstration: AWS Security Hub, Amazon GuardDuty, AWS Config und Amazon Inspector
◊

- **Konfigurationsmanagement**
  - ◊ Einführung in den Konfigurationsmanagementprozess
  - ◊ AWS-Services und Tools für Konfigurationsmanagement
  - ◊ Hands-on Lab: Durchführung von Blue/Green-Deployments mit CI/CD-Pipelines und Amazon ECS
  - ◊
- **Observability**
  - ◊ Einführung in Observability
  - ◊ AWS-Tools zur Unterstützung von Observability
  - ◊ Hands-on Lab: Einsatz von AWS DevOps-Tools zur CI/CD-Pipeline-Automatisierung
  - ◊
- **Referenzarchitektur (optional)**
  - ◊ Referenzarchitekturen
  - ◊
- **Kurszusammenfassung**
  - ◊ Komponenten der DevOps-Praxis
  - ◊ Rückblick auf die CI/CD-Pipeline
  - ◊ AWS-Zertifizierung
  - ◊