

AW330 Developing Generative AI Applications on AWS

Kurzbeschreibung:

Teilnehmer erhalten eine Einführung in generative KI mit Fokus auf Amazon Bedrock und LangChain. Behandelt werden die Grundlagen des Prompt Engineerings, Architekturkonzepte zur Entwicklung generativer Anwendungen sowie die Planung von KI-Projekten. Präsentationen, Demonstrationen und Gruppenübungen sichern den Praxistransfer.

Zielgruppe:

Dieser Kurs **AW330 Developing Generative AI Applications on AWS** richtet sich an:

- Softwareentwickler, die große Sprachmodelle nutzen möchten, ohne sie feinabzustimmen

Voraussetzungen:

Um an dem Kurs **AW330 Developing Generative AI Applications on AWS** bei qSkills teilnehmen zu können, sollten Sie das folgende AWS-Training besucht haben:

- [AW110 AWS Technical Essentials](#)

Darüber hinaus sollten Sie folgende Voraussetzungen erfüllen:

- Kenntnisse auf mittlerem Niveau in Python

Sonstiges:

Dauer: 2 Tage

Preis: 1250 Euro plus Mwst.

Ziele:

In diesem Kurs **AW330 Developing Generative AI Applications on AWS** lernen Sie:

- Generative KI zu beschreiben und ihre Einordnung in maschinelles Lernen zu erläutern
- Die Bedeutung von generativer KI zu definieren und potenzielle Risiken und Vorteile zu erklären
- Geschäftlichen Mehrwert aus Use Cases generativer KI zu identifizieren
- Die technischen Grundlagen und Schlüsselbegriffe generativer KI zu diskutieren
- Die Schritte zur Planung eines generativen KI-Projekts zu erläutern
- Einige Risiken und Gegenmaßnahmen beim Einsatz generativer KI zu identifizieren
- Zu verstehen, wie Amazon Bedrock funktioniert
- Sich mit grundlegenden Konzepten von Amazon Bedrock vertraut zu machen
- Die Vorteile von Amazon Bedrock zu erkennen
- Typische Anwendungsfälle für Amazon Bedrock aufzulisten
- Die typische Architektur einer Amazon-Bedrock-Lösung zu beschreiben
- Die Kostenstruktur von Amazon Bedrock zu verstehen
- Eine Demonstration von Amazon Bedrock in der AWS Management Console umzusetzen

- Prompt Engineering zu definieren und allgemeine Best Practices im Umgang mit FMs anzuwenden
- Die grundlegenden Arten von Prompt-Techniken zu identifizieren, darunter Zero-Shot und Few-Shot Learning
- Erweiterte Prompt-Techniken bei Bedarf für eigene Use Cases anzuwenden
- Die für spezifische Modelle am besten geeigneten Prompt-Techniken zu identifizieren
- Potenziellen Missbrauch von Prompts zu identifizieren
- Potenzielle Verzerrungen in FM-Antworten zu analysieren und Prompts zur Minderung dieser Verzerrungen zu entwerfen
- Die Komponenten einer generativen KI-Anwendung zu identifizieren und wie ein Foundation Model (FM) angepasst wird
- Amazon Bedrock Foundation Models, Inferenzparameter und wichtige Amazon-Bedrock-APIs zu beschreiben
- AWS-Angebote zu identifizieren, die beim Monitoring, bei der Absicherung und bei der Governance von Amazon-Bedrock-Anwendungen unterstützen
- Zu beschreiben, wie LangChain mit Large Language Models (LLMs), Prompt Templates, Chains, Chat Models, Text Embedding Models, Document Loaders, Retrievers und Agents für Amazon Bedrock integriert wird
- Architekturkonzepte zu beschreiben, die mit Amazon Bedrock zur Entwicklung generativer KI-Anwendungen implementiert werden können
- Die Konzepte anzuwenden, um Beispiel-Use-Cases zu entwickeln und zu testen, die verschiedene Amazon-Bedrock-Modelle, LangChain und den Retrieval-Augmented-Generation-Ansatz (RAG) nutzen

Inhalte/Agenda:

- **♦ Einführung in generative KI – Art of the Possible**
 - ♦ Überblick über ML
 - ♦ Grundlagen der generativen KI
 - ♦ Anwendungsfälle generativer KI
 - ♦ Generative KI in der Praxis
 - ♦ Risiken und Vorteile
- **♦**
- **♦ Planung eines generativen KI-Projekts**
 - ♦ Grundlagen generativer KI
 - ♦ Generative KI in der Praxis
 - ♦ Kontext generativer KI
 - ♦ Schritte zur Planung eines generativen KI-Projekts
 - ♦ Risiken und Gegenmaßnahmen
- **♦**
- **♦ Einstieg in Amazon Bedrock**
 - ♦ Einführung in Amazon Bedrock
 - ♦ Architektur und Anwendungsfälle
 - ♦ Nutzung von Amazon Bedrock
 - ♦ Demonstration: Zugriff auf Amazon Bedrock einrichten und Playgrounds nutzen
- **♦**
- **♦ Grundlagen des Prompt Engineerings**
 - ♦ Grundlagen von Foundation Models
 - ♦ Grundlagen des Prompt Engineerings
 - ♦ Grundlegende Prompt-Techniken
 - ♦ Erweiterte Prompt-Techniken
 - ♦ Demonstration: Feinabstimmung eines einfachen Text-Prompts
 - ♦ Modellspezifische Prompt-Techniken
 - ♦ Umgang mit fehlerhaften Prompts
 - ♦ Minderung von Verzerrungen
 - ♦ Demonstration: Bias-Mitigation bei Bildern
- **♦**
- **♦ Komponenten generativer KI-Anwendungen mit Amazon Bedrock**
 - ♦ Anwendungen und Use Cases
 - ♦ Überblick über Komponenten generativer KI-Anwendungen
 - ♦ Foundation Models und die FM-Schnittstelle
 - ♦ Arbeit mit Datasets und Embeddings
 - ♦ Demonstration: Wort-Embeddings
 - ♦ Weitere Anwendungsbestandteile
 - ♦ RAG
 - ♦ Modell-Fine-Tuning
 - ♦ Absicherung generativer KI-Anwendungen
 - ♦ Architektur generativer KI-Anwendungen
- **♦**
- **♦ Amazon Bedrock Foundation Models**
 - ♦ Einführung in Amazon Bedrock Foundation Models
 - ♦ Nutzung von Amazon Bedrock FMs zur Inferenz
 - ♦ Amazon Bedrock Methoden
 - ♦ Datenschutz und Auditierbarkeit
- **♦**
- **♦ Lab: Aufruf eines Amazon-Bedrock-Modells zur Textgenerierung mittels Zero-Shot-Prompt**
- **♦**
- **♦ LangChain**
 - ♦ Optimierung der LLM-Leistung
 - ♦ Integration von AWS und LangChain
 - ♦ Nutzung von Modellen mit LangChain
 - ♦ Aufbau von Prompts
 - ♦ Strukturierung von Dokumenten mit Indizes
 - ♦ Speicherung und Abruf von Daten mit Memory
 - ♦ Nutzung von Chains zur Sequenzierung von Komponenten
 - ♦ Verwaltung externer Ressourcen mit LangChain Agents
- **♦**
- **♦ Architekturkonzepte**
 - ♦

- ◇ Einführung in Architekturkonzepte
- ◇ Textzusammenfassung
- ◇ Lab: Nutzung von Amazon Titan Text Premier zur Zusammenfassung kurzer Texte
- ◇ Lab: Lange Texte mit Amazon Titan zusammenfassen
- ◇ Question Answering
- ◇ Lab: Nutzung von Amazon Bedrock für Question Answering
- ◇ Chatbots
 - ◇ · Lab: Chatbot entwickeln
- ◇ Codegenerierung
 - ◇ · Lab: Nutzung von Amazon-Bedrock-Modellen zur Codegenerierung
- ◇ LangChain und Agents für Amazon Bedrock
 - ◇ · Lab: Konversationsanwendungen mit der Converse API entwickeln