

Applikationen im Cloud-Zeitalter

Container, Cloud & Microservices

Themen wie Cloud und Containerization sind überall präsent. Für viele Anwendungsentwickler oder -designer ist heute jedoch immer noch schwer abzuschätzen, welche Auswirkungen es haben kann, wenn eine Applikation in eine Cloud-Umgebung migriert wird.

Häufig geht der Umzug in die Cloud sogar über eine einfache Migration hinaus und es wird eine Modernisierung der Anwendung und das Überführen in eine Microservice-Architektur angestrebt. Im Laufe dieses Kurses werden die Anforderungen und Problematiken einer Cloud- und Containerplattform im Zusammenhang mit Applikationen detailliert aufgezeigt und besprochen.

Welche Möglichkeiten zur Automation bieten Container? Wo sind die Grenzen? Von welchen Gewohnheiten muss man sich mitunter auch trennen? Was haben Abkürzungen wie DevOps oder CI/CD damit zu tun?

Dabei werden die Konzepte Cloud und Container aus technischer Sicht beleuchtet, sowie deren Zusammenhang zur Applikation. Ziel ist es, ein Verständnis zu schaffen, welche Auswirkungen oder Änderungsanforderungen auf eine Applikation zukommen, wenn diese in eine Cloud- oder Containerplattform überführt wird.

Kursinhalt

- Grundlagen zu Cloud Computing
- Mögliche Migrationsszenarien in die Cloud (Refactoring, Lift & Shift, Replace)
- Aufbau moderner Anwendungen (12-Factor-App, GIT, Microservices, ...)
- Grundlagen Containerization am Beispiel von Docker
- Container-Images und Dockerfile
- Ressourcen-Management und Möglichkeiten der Automation
- Container-Management mit Kubernetes
- Deployments und Rolling-Updates
- Entwickeln und Ausrollen von Applikationen: Extreme Programming, Test-Driven Development (TDD) und Behavior-Driven Development, CI/CD, DevOps, ...
- Ausblick: DevOps-Tools und Orchestrierung in der Cloud

E-Book Das ausführliche deutschsprachige digitale Unterlagenpaket, bestehend aus PDF und E-Book, ist im Kurspreis enthalten.

Zielgruppe

Der Kurs richtet sich an alle, die einen Einstieg in das Thema Applikationen in der Cloud suchen, ein gewisses technisches Verständnis mitbringen und sich nicht vor Neuerungen scheuen.

Personen, die im Bereich Anwendungsentwicklung und -Design arbeiten, die mit Plattform-Architekturen betraut sind oder im PreSales mit Applikationsmigrationen, Entwicklung oder Design zu tun haben, liefert der Kurs ein stabiles Know-how-Fundament und einen fundierten Einblick in den Zusammenhang von Applikationen, Containern und Entwicklung.

Voraussetzungen

Gefragt ist die Bereitschaft zur Auseinandersetzung mit konzeptionellen und technischen Themen im Cloud-Kontext. Tiefes technisches Wissen oder Programmierkenntnisse sind nicht erforderlich. Allerdings sollten Sie über grundlegendes IT-Know-How verfügen und neuen Themen gegenüber aufgeschlossen sein.

Alternativen

Buchen Sie diesen Kurs zusammen mit *Software Engineering im Cloud-Zeitalter – Microservices in der Praxis* als *PowerPackage Cloud-Applikationen* zum vergünstigten Preis von **€ 2.795,-** statt insgesamt € 3.790,- bei Einzelbuchung der beiden Kurse.

Dieser Kurs im Web



Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link: www.experteach.de/go/ADIC

Vormerkung

Sie können auf unserer Website einen Platz kostenlos und unverbindlich für 7 Tage reservieren. Dies geht auch telefonisch unter 06074 4868-0.

Garantierte Kurstermine

Für Ihre Planungssicherheit bieten wir stets eine große Auswahl garantierter Kurstermine an.

Ihr Kurs maßgeschneidert

Diesen Kurs können wir für Ihr Projekt exakt an Ihre Anforderungen anpassen.

Training		Preise zzgl. MwSt.	
Termine in Deutschland	3 Tage	€ 1.995,-	
Online Training	3 Tage	€ 1.995,-	
Termin/Kursort	Kurssprache Deutsch		
27.10.-29.10.25	Frankfurt	20.04.-22.04.26	Frankfurt
27.10.-29.10.25	Online	20.04.-22.04.26	Online

Stand 23.05.2025



EXPERTeach



Inhaltsverzeichnis

Applikationen im Cloud-Zeitalter – Container, Cloud & Microservices

1 Der Trend: Cloud Computing	4 Docker im Detail	6.9 Das Spotify Modell: Squad, Chapter, Tribe
1.1 IT im Wandel	4.1 Docker im Detail	6.10 Scaled Agile Framework (SAFe)
1.2 Treiber für die Cloud	4.1.1 Container Execution Environments	6.11 Bimodale IT
1.2.1 Verfügbarkeit des Business	4.2 Bestandteile von Docker	
1.2.2 Von fixen zu variablen Kosten	4.3 Automatisierung mit Docker	
1.2.3 Agile Infrastruktur	4.4 Container-Security	
1.2.4 Technologisch immer aktuell		
1.2.5 Sicherheit und Compliance	5 Kubernetes	
1.3 Typische Herausforderungen und Einwände	5.1 Kubernetes	
1.3.1 Herausforderungen beim Cloud Computing für die Provider	5.1.1 Kubernetes-Namespace	
1.3.2 Faktoren für die Kundenzufriedenheit	5.1.2 Kubernetes-Pod	
1.4 Virtualization – Enabler für Cloud Computing	5.1.3 Kubernetes-Deployment	
1.5 Definition: Cloud Computing	5.1.4 Kubernetes-Services	
1.5.1 Service-Modelle des Cloud Computings	5.2 Kubernetes und Netzwerk	
1.5.2 Die verschiedenen Cloud-Varianten (Private Cloud, Public Cloud, ...)	5.3 Orchestrierung mit Container	
1.5.3 Multi-Cloud	5.3.1 Swarm	
1.5.4 Eigenschaften der Hyperscaler	5.3.2 Stateless Applications	
1.5.5 Shared Responsibility	5.3.3 Web Services und Port-Bindung	
	5.3.4 Microservices	
2 Applikationen in der Cloud	6 Betriebskonzepte und agile Methoden	
2.1 Applikationen in der Cloud	6.1 Klassiker: Plan/Build/Run und ITIL	
2.1.1 Aufbau von Applikationen	6.1.1 Auswirkungen auf die Prozesse	
2.1.2 Eignungsprüfung für Applikationen in der Cloud	6.1.2 IT-Kennzahlen	
2.2 Cloud Native Applications	6.2 Virtualisierung und Rollen	
2.2.1 12-Factor-App	6.2.1 Klassische Rollen in der IT	
2.2.2 Versionsverwaltung	6.2.2 Auswirkung von Eigenschaften virtualisierter Umgebungen	
2.2.3 Saubere Trennung des eigentlichen Codes von anderem	6.2.3 Neue Rollen	
2.2.4 Build, Release, Run	6.2.4 Auswirkungen von Microservices und Containerization auf den IT-Betrieb	
2.2.5 Stateless Applications	6.3 Agile Methoden	
2.2.6 Web Services und Port-Bindung	6.4 Extreme Programming (XP)	
2.3 Microservices	6.4.1 Test-Driven Development (TDD)	
2.3.1 Monolith vs. Microservices	6.4.2 Refactoring (Design Improvement)	
2.4 Horizontale Skalierbarkeit	6.4.3 Continuous Integration	
2.4.1 Weitere Faktoren	6.4.4 Behavior-Driven Development (BDD)	
2.5 Das Chaos-Monkey-Prinzip	6.4.5 Planning in Short Cycles	
2.6 Pets vs. Cattle	6.5 Lean Development	
	6.5.1 MVP	
3 Einführung in die Container-Virtualisierung	6.6 DevOps	
3.1 Evolutionsstufen der Virtualisierung	6.6.1 Continuous Delivery	
3.2 Container-Virtualisierung	6.6.2 CI/CD	
3.2.1 Linux Containers (LXC)	6.7 Kanban	
3.2.2 LXD (Linux Container Hypervisor)	6.8 Scrum	
3.3 Docker		

