Designing and Implementing Microsoft DevOps Solutions (AZ-400T00)

Dieser Kurs behandelt Design und Implementierung von DevOps-Prozessen und -Praktiken. Die Teilnehmer lernen, für DevOps zu planen, Quellensteuerung zu verwenden, Git für ein Unternehmen zu skalieren, Artefakte zu konsolidieren, eine Strategie zur Verwaltung von Abhängigkeiten zu entwerfen, kontinuierliche Integration und eine Containerstrategie zu implementieren, eine Releasestrategie und -verwaltung einzurichten, ein Bereitstellungsmuster zu implementieren und Feedbackmechanismen zu optimieren. Darüber hinaus dient der Kurs zur optimalen Vorbereitung auf das Examen AZ-400: Designing and Implementing Microsoft DevOps Solutions für die Zertifizierung als Microsoft Certified: DevOps Engineer Expert.

INHALTE

Planung für DevOps

- Transformationsplanung
- Projektauswahl
- Teamstrukturen

Einführung in Quellensteuerung

- Was ist Quellensteuerung?
- Vorteile der Quellensteuerung
- Arten von Quellensteuerungssystemen
- Einführung in Azure Repos
- Einführung in GitHub
- Migration von Team Foundation Version Control (TFVC) auf Git in Azure Repos
- Authentifizierung an Git Repos

Skalierung von Git für Unternehmens-DevOps

- Strukturierung des Git Repo
- Git-Branching-Workflows
- Zusammenarbeit mit Pull Requests in Azure Repos
- GitHooks
- Fördern von internen Quellen

Konsolidierung von Artefakten und Design einer Strategie zur Abhängigkeitsverwaltung

- Abhängigkeiten bei der Paketierung
- Empfehlungen für Tools und Praktiken zur Artefaktverwaltung

PREIS P. P.

€ 2690,- (zzgl. MwSt.)

DAUER

4 Tage (09:00 - 17:00 Uhr)

SIE HABEN FRAGEN?

+43 50 4510-0

E-Mail Anfrage: office@tectrain.at

https://dev.tectrain.at/seminare/programmierung/microsoft-technisch/designing-and-implementing-microsoft-devops-solutions

- Zusammenfassen gängiger Pakete, um deren gemeinsame Nutzung und Wiederverwendung zu ermöglichen
- Migration und Konsolidierung von Artefakten
- Migration und Integration von Quellensteuerung

Implementierung der kontinuierlichen Integration mit Azure-Pipelines

- Das Konzept von Pipelines in DevOps
- Azure-Pipelines
- Evaluierung der Nutzung von gehosteten vs. privaten Agenten
- Agentenpools
- Pipelines und gleichzeitige Zugriffe
- Azure DevOps und Open-Source-Projekte (öffentliche Projekte)
- Azure-Pipelines-YAML vs. Visual Designer
- Kontinuierliche Integration im Überblick
- Implementierung einer Build-Strategie
- Integration mit Azure-Pipelines
- Integration externer Quellensteuerung mit Azure-Pipelines
- Einrichtung privater Agenten
- Analyse und Integration von mehrstufigen Docker-Builds

Verwaltung von Anwendungskonfiguration und -geheimnissen

- Einführung in die Sicherheit
- Implementierung eines sicheren und compliancegerechten Entwicklungsprozesses
- Überdenken von Anwendungskonfigurationsdaten
- Verwaltung von Geheimnissen, Tokens und Zertifikaten
- Implementierung von Tools für die Verwaltung von Sicherheit und Compliance in einer Pipeline

Planung für Qualität und Sicherheit

- Planung einer Qualitätsstrategie
- Planung einer sicheren Entwicklung

Verwaltung von Codequalität und Sicherheitsrichtlinien

- Verwaltung der Codequalität
- Verwaltung von Sicherheitsrichtlinien mit Open Source, OWASP und WhiteSource Bolt

Implementierung einer Container-Build-Strategie

- Unterschiede zwischen Containern und virtuellen Maschinen
- Verwendung von Containern durch Microservices
- Implementierung von Containern mit Docker

Verwaltung von Versionierung, Sicherheit und Compliance für Artefakte

- Paketsicherheit
- Open-Source-Software
- Integration von Lizenz- und Schwachstellenscans
- Implementieren einer Versionierungsstrategie (Git-Version)

Design einer Releasestrategie

- Einführung in die kontinuierliche Bereitstellung
- Empfehlungen zur Releasestrategie
- Aufbau einer Releasepipeline mit hoher Qualität
- Auswahl eines Bereitstellungsmusters
- Auswahl des passenden Werkzeugs für die Releaseverwaltung

Einrichtung eines Releaseverwaltungsworkflows

- Erstellen einer Releasepipeline
- Bereitstellung und Konfiguration von Umgebungen
- Verwalten und Modularisieren von Aufgaben und Vorlagen
- Integration von Geheimnissen mit der Releasepipeline
- Konfiguration der automatisierten Integration und funktionale Testautomatisierung
- Automatisierung der Untersuchung des Systemzustands

Implementierung eines geeigneten Bereitstellungsmusters

- Einführung in Bereitstellungsmuster
- Implementierung von Blue-Green-Bereitstellung
- Feature Toggles
- Canary-Releases
- Dark Launching
- AB-Tests
- Bereitstellung als Progressive Exposure

Implementierung von Prozessen, um Systemfeedback an Entwicklungsteams weiterzuleiten

- Implementierung von Tools, um System- und Featurenutzung nachzuverfolgen
- Implementierung des Routings für Berichtsdaten über Abstürze mobiler Anwendungen
- Entwicklung von Überwachungs- und Statusdashboards
- Integration und Konfiguration von Ticketingsystemen

Implementierung einer mobilen DevOps-Strategie

- Einführung in Mobile DevOps

- Einführung in Visual Studio App Center
- Verwaltung mobiler Gerätesätze und Verteilungsgruppen
- Verwaltung von UI-Testgerätesätzen
- Vorhalten von Testergeräten für die Bereitstellung
- Erstellen von öffentlichen und privaten Verteilungsgruppen

Azure-Tools für Infrastruktur und Konfiguration

- Infrastruktur als Code- und Konfigurationsverwaltung
- Erstellen von Azure-Ressourcen mithilfe von ARM-Vorlagen
- Erstellen von Azure-Ressourcen mithilfe der Azure CLI
- Erstellen von Azure-Ressourcen mithilfe von Azure PowerShell
- Zusätzliche Automatisierungswerkzeuge

Azure-Bereitstellungsmodelle und -dienste

- Bereitstellungsmodule und -optionen
- Azure-Infrastructure-as-a-Service (-IaaS)-Dienste
- Azure-Automatisierung mit DevOps
- Desired State Configuration (DSC)
- Azure-Platform-as-a-Service (-PaaS)-Dienste
- Azure Service Fabric

Erstellen und Verwalten einer Kubernetes-Service-Infrastruktur

- Bereitstellung und Konfiguration eines verwalteten Kubernetes-Clusters

Mit Azure verfügbare Drittanbieter- und Open-Source-Tools

- Chef
- Puppet
- Ansible
- Terraform

Implementierung von Compliance und Sicherheit in der Infrastruktur

- Sicherheits- und Complianceprinzipien mit DevOps
- Azure Security Center

Empfehlen und Entwerfen von Systemfeedbackmechanismen

- Die innere Schleife
- Continuous Experimentation
- Designpraktiken, um die Zufriedenheit der Endanwender zu messen
- Designprozesse, um Benutzerfeedback einzuholen und zu

analysieren
- Designprozesse, um Anwendungsanalysen zu automatisieren

Optimierung von Feedbackmechanismen

- Site Reliability Engineering
- Analyse von Telemetrie, um eine Grundlinie zu etablieren
- Laufendes Tuning, um bedeutungslose und nicht umsetzbare Alarme zu reduzieren
- Analyse von Alarmen, um eine Grundlinie zu etablieren
- Blameless PostMortems und Just Culture

ZIELGRUPPE

Azure DevOps Engineers

VORAUSSETZUNGEN

Besuch des Kurses AZ-900T00: Microsoft Azure Fundamentals oder vergleichbare Kenntnisse. Kenntnisse zu Versionskontrolle, agiler Softwareentwicklung und wichtigen Prinzipien der Softwareentwicklung.

ZERTIFIZIERUNG

AZ-400: Designing and Implementing Microsoft DevOps Solutions

Opt. Zertifizierungsgebühr: € 165,- zzgl. MwSt.

ABSCHLUSS

Nach Seminarabschluss erhalten Sie ein tecTrain-Teilnahmezertifikat.