Enterprise Computing Solutions - Education Services

TRAINING OFFERING

Sie erreichen uns hier

Freistädterstraße 236, A-4040 Linz

Email: education.ecs.at@arrow.com
Phone: +43 1 370 94 40 - 34

Designing and Implementing Microsoft DevOps Solutions

CODE:	LÄNGE:	PREIS:
MCS_AZ-400T00	40 Hours (5 Tage)	€2,875.00

Description

Dieser Kurs vermittelt die Kenntnisse und Fähigkeiten zur Implementierung, Umsetzung von DevOps-Prozessen und Praktiken. Entwerfen Sie dabei Release-Strategien und richten Sie Release-Management-Workflows ein. Außerdem erwerben Sie Fähigkeiten zur Implementierung von Dependency Management Strategien und zur Bereitstellung von Anwendungsinfrastruktur in DevOps-Pipelines mit Rücksicht auf Compliance und Sicherheitsanforderungen.
Des Weiteren lernen Sie in diesem Seminar wie Sie kontinuierliches Feedback mithilfe von System-Feedback-Mechanismen implementieren können. Schließlich können Sie nach diesem Training eine DevOps-Strategie entwerfen inkl. der Planung einer Transformation und des Projektes sowie die Erstellung von effizienten Teamstrukturen.

Lernziel

Nach Abschluss dieses Seminars haben die Teilnehmer Wissen zu folgenden Themen:
- Planning for DevOps
- Getting Started with Source Control
- Managing Technical Debt
- Working with Git for Enterprise DevOps
- Configuring Azure Pipelines
- Implementing Continuous Integration using Azure Pipelines
- Managing Application Configuration and Secrets
- Implementing Continuous Integration with GitHub Actions
- Designing and Implementing a Dependency Management Strategy
- Designing a Release Strategy
- Implementing Continuous Deployment using Azure Pipelines
- Implementing an Appropriate Deployment Pattern
- Managing Infrastructure and Configuration using Azure Tools
- Third Party Infrastructure as Code Tools Available with Azure
- Managing Containers using Docker
- Creating and Managing Kubernetes Service Infrastructure
- Implementing Feedback for Development Teams
- Implementing System Feedback Mechanisms
- Implementing Security in DevOps Projects
- Validating Code Bases for Compliance

Zielgruppe

Dieses Seminar richtet sich an:
- Teilnehmer, die DevOps Projekte und Prozesse planen und implementieren möchten
- Teilnehmer, die eine Anwendungsinfrastruktur implementieren und diese managen sowie konfigurieren möchten
- Teilnehmer, die an kontinuierlichen DevOps Integrationsprozessen interessiert sind
- Teilnehmer, die an der Zertifizierungsprüfung für Microsoft Azure DevOps Engineer interessiert sind

Voraussetzungen

Für dieses Seminar werden folgende Vorkenntnisse empfohlen:

Fundierte Kenntnisse über Azure, Erfahrung mit virtuellen Maschinen und Containern und einige Erfahrungen mit Automatisierung und Skripting
Grundlegende Kenntnisse über Versionskontrolle, Agile Softwareentwicklung und grundlegende Prinzipien der Softwareentwicklung
Erfahrung in einer Softwareentwicklungs- oder Betriebsumgebung mit Windows oder Linux wäre hilfreich, ist aber nicht zwingend erforderlich
Die Teilnehmer sollten auch Kenntnisse über allgemeine Prozesse der Anwendungsentwicklung und -bereitstellung haben
Außerdem wird empfohlen, dass Sie über Erfahrung in der Arbeit mit einer IDE sowie über einige Kenntnisse zum Azure-Portal verfügen. Teilnehmer, die keinen technischen Hintergrund in diesen Technologien haben, aber neugierig auf DevOps-Praktiken als Kulturwandel sind, sollten jedoch in der Lage sein, die prozeduralen und expositorischen Erklärungen der kontinuierlichen Integration unabhängig davon zu verfolgen.
In diesem Seminar erlernen Sie notwendige Vorkenntnisse für das DevOps Seminar: Developing Solutions für Microsoft Azure

Inhalt

Transformation Planning
Project Selection
Team Structures
Planning for DevOps Migrating to Azure DevOps Getting Started with Source Control
What is Source Control
Benefits of Source Control
Types of Source Control Systems
Introduction to Azure Repos
Introduction to GitHub
Migrating from Team Foundation Version Control (TFVC) to Git in Azure Repos Managing Technical Debt
Identifying Technical Debt
Knowledge Sharing within Teams
Modernizing Development Environments with Codespaces Working with Git for Enterprise DevOps
How to Structure Your Git Repo
Git Branching Workflows
Collaborating with Pull Requests in Azure Repos
Why Care About Git Hooks
Fostering Inner Source
Managing Git Repositories Configuring Azure Pipelines
The Concept of Pipelines in DevOps
Azure Pipelines
Evaluate use of Hosted versus Self-Hosted Agents
Agent Pools
Pipelines and Concurrency
Azure DevOps and Open-Source Projects (Public Projects)
Azure Pipelines YAML versus Visual Designer Implementing Continuous Integration using Azure Pipelines
Continuous Integration Overview
Implementing a Build Strategy
Integration with Azure Pipelines
Integrating External Source Control with Azure Pipelines
Set Up Self-Hosted Agents Managing Application Configuration and Secrets
Introduction to Security
Implement a Secure Development Process
Rethinking Application Configuration Data
Manage Secrets, Tokens, and Certificates
Integrating with Identity Management Systems
Implementing Application Configuration Implementing Continuous Integration with GitHub Actions
GitHub Actions
Continuous Integration with GitHub Actions
Securing Secrets for GitHub Actions Designing and Implementing a Dependency Management Strategy
Packaging Dependencies
Package Management Introduction to Continuous Delivery
Migrating and Consolidating Artifacts Release Strategy Recommendations
Package Security Building a High-Quality Release pipeline
Implementing a Versioning Strategy Designing a Release Strategy Choosing the Right Release Management Tool
Create a Release Pipeline
Provision and Configure Environments
Manage and Modularize Tasks and Templates
Configure Automated Integration and Functional Test Automation
Implementing Continuous Deployment using Azure Pipelines Automate Inspection of Health

Introduction to Deployment Patterns
Implement Blue Green Deployment
Feature Toggles
Canary Releases
Dark Launching
AB Testing
Implementing an Appropriate Deployment Pattern
Progressive Exposure Deployment
Infrastructure as Code and Configuration Management
Create Azure Resources using ARM Templates
Create Azure Resources using Azure CLI
Azure Automation with DevOps
Managing Infrastructure and Configuration using Azure Tools
Desired State Configuration (DSC)
Chef
Puppet
Ansible
Third Party Infrastructure as Code Tools Available with Azure
Terraform Managing Containers using Docker
Implementing a Container Build Strategy
Implementing Docker Multi-Stage Builds
Creating and Managing Kubernetes Service Infrastructure
Azure Kubernetes Service
Kubernetes Tooling
Integrating AKS with Pipelines
Implementing Feedback for Development Teams
Implement Tools to Track System Usage, Feature Usage, and Flow
Implement Routing for Mobile Application Crash Report Data
Develop Monitoring and Status Dashboards
Integrate and Configure Ticketing Systems
Implementing System Feedback Mechanisms
Site Reliability Engineering
Design Practices to Measure End-User Satisfaction
Design Processes to Capture and Analyze User Feedback
Design Processes to Automate Application Analytics
Managing Alerts
Security in the Pipeline
Blameless Retrospectives and a Just Culture
Implementing Security in DevOps Projects
Azure Security Center
Open-Source Software
Managing Security and Compliance Policies
Validating Code Bases for Compliance
Integrating License and Vulnerability Scans

Test und Zertifizierung

Dieses Seminar basiert auf der originalen Microsoft MOC Unterlage AZ-400 und bereitet auf die dementsprechende Prüfung vor.
Mögliche Zertifizierung: Microsoft Azure DevOps Engineer
Wichtige Information
Dieses Seminar behandelt prüfungsrelevante Themen zum Microsoft Examen: AZ-400 Microsoft Azure DevOps Solutions
Mögliche Zertifizierung: Microsoft Azure DevOps Engineer

Kurstermine

Auf Anfrage. Bitte kontaktieren Sie uns

Zusätzliche Information

Diese Schulung ist auch als Vor-Ort-Schulung verfügbar. Bitte kontaktieren Sie uns, um mehr zu erfahren.