

AZ-400T00

Designing and Implementing Microsoft DevOps solutions

Dieser Kurs vermittelt die Kenntnisse und Fähigkeiten zur Entwicklung und Umsetzung von DevOps-Prozessen und -praktiken. Die Teilnehmer*innen lernen, wie sie DevOps planen, Quellcodeverwaltung verwenden, Git für ein Unternehmen skalieren, Artefakte konsolidieren, eine Strategie für das Abhängigkeitsmanagement entwerfen, Geheimnisse verwalten, Continuous-Integration-Ansätze und eine Strategie zum Erstellen von Containern implementieren, eine Releasestrategie entwerfen, einen Releaseverwaltungsworkflow einrichten, ein Bereitstellungsmuster implementieren und Feedbackmechanismen optimieren.

Kursinhalt

- Einführung in DevOps
- Auswählen des richtigen Projekts
- Beschreiben von Teamstrukturen
- Auswählen der DevOps-Tools
- Agile-Planung mit GitHub-Projekten und Azure Boards
- Einführung in die Quellcodeverwaltung
- Beschreiben von Typen von Quellcodeverwaltungssystemen
- Arbeiten mit Azure Repos und GitHub

E-Book Die originalen Microsoft-Kursunterlagen werden Ihnen online zur Verfügung gestellt.

Zielgruppe

Die Teilnehmer dieses Kurses sind daran interessiert, DevOps-Prozesse zu entwerfen und zu implementieren oder die Zertifizierungsprüfung für Microsoft Azure DevOps-Lösungen abzulegen.

Voraussetzungen

Erfolgreiche Kursteilnehmer verfügen über die folgenden Vorkenntnisse und Kenntnisse:

- Cloud Computing-Konzepte, einschließlich eines Verständnisses von PaaS-, SaaS- und IaaS-Implementierungen.
- Erfahrung in Bezug auf Azure-Verwaltung und Azure-Entwicklung mit nachgewiesenen Kenntnissen in mindestens einem dieser Bereiche
- Kenntnisse in Bezug auf Versionskontrolle, agile Softwareentwicklung und die wichtigsten Prinzipien der Softwareentwicklung Es wäre hilfreich, wenn Sie Erfahrung in einem Unternehmen hätten, das Software liefert.

Folgende Kurse bieten sich zur Vorbereitung an:

Wenn Sie noch nicht mit Azure und Cloud Computing vertraut sind:

AZ-900: Azure Fundamentals

Wenn Sie noch nicht mit der Azure-Verwaltung vertraut sind:

AZ-104: Microsoft Azure Administrator

Wenn Sie noch nicht mit Azure Developer vertraut sind

AZ-204: Entwickeln von Lösungen für Microsoft Azure

Kursziel

Der Kurs unterstützt die Teilnehmer auf die Vorbereitung zum Examen AZ-400, welches für die Zertifizierung "Microsoft Certified: DevOps Engineer Expert" vorausgesetzt wird.

Stand 09.03.2025

Dieser Kurs im Web



Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link: www.experteach.ch/go/MZ40

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		5 Tage CHF 3.075,-	
Online Training		5 Tage CHF 3.075,-	
Termin/Kursort		Kurssprache Deutsch	
02.06.-06.06.25	ONOnline	17.11.-21.11.25	HYFrankfurt
25.08.-29.08.25	HYFrankfurt	17.11.-21.11.25	HYOnline
25.08.-29.08.25	HYOnline		

Inhaltsverzeichnis

AZ-400T00 – Designing and Implementing Microsoft DevOps solutions

Module 1: Get started on a DevOps transformation journey

Lessons

Introduction to DevOps

Choose the right project

Describe team structures

Choose the DevOps tools

Plan Agile with GitHub Projects and Azure Boards

Introduction to source control

Describe types of source control systems

Work with Azure Repos and GitHub

Lab : Agile planning and portfolio management with Azure Boards

Lab : Version controlling with Git in Azure Repos

After completing this module, students will be able to:

Understand what DevOps is and the steps to accomplish it

Identify teams to implement the process

Plan for the transformation with shared goals and timelines

Plan and define timelines for goals

Understand different projects and systems to guide the journey

Select a project to start the DevOps transformation

Identify groups to minimize initial resistance

Identify project metrics and Key Performance Indicators (KPI's)

Understand agile practices and principles of agile development

Create a team and agile organizational structure

Module 2: Development for enterprise DevOps

Lessons

Structure your Git Repo

Manage Git branches and workflows

Collaborate with pull requests in Azure Repos

Explore Git hooks

Plan foster inner source

Manage Git repositories

Identify technical debt

Lab : Version controlling with Git in Azure Repos

After completing this module, students will be able to:

Understand Git repositories

Implement mono repo or multiple repos

Explain how to structure Git Repos

Implement a change log

Describe Git branching workflows

Implement feature branches

Implement GitFlow

Fork a repo

Leverage pull requests for collaboration and code reviews

Give feedback using pull requests

Module 3: Implement CI with Azure Pipelines and GitHub Actions

Lessons

Explore Azure Pipelines

Manage Azure Pipeline agents and pools

Describe pipelines and concurrency

Explore Continuous integration

Implement a pipeline strategy

Integrate with Azure Pipelines

Introduction to GitHub Actions

Learn continuous integration with GitHub Actions

Design a container build strategy

Lab : Configuring agent pools and understanding pipeline styles

Lab : Enabling continuous integration with Azure Pipelines

Lab : Integrating external source control with Azure Pipelines

Lab : Implementing GitHub Actions by using DevOps Starter

Lab : Deploying Docker Containers to Azure App Service web apps

After completing this module, students will be able to:

Describe Azure Pipelines

Explain the role of Azure Pipelines and its components

Decide Pipeline automation responsibility

Understand Azure Pipeline key terms

Choose between Microsoft-hosted and self-hosted agents

Install and configure Azure pipelines Agents

Configure agent pools

Make the agents and pools secure

Use and estimate parallel jobs

Module 4: Design and implement a release strategy

Lessons

Introduction to continuous delivery

Create a release pipeline

Explore release strategy recommendations

Provision and test environments

Manage and modularize tasks and templates

Automate inspection of health

Lab : Creating a release dashboard

Lab : Controlling deployments using Release Gates

After completing this module, students will be able to:

Explain continuous delivery (CD)

Implement continuous delivery in your development cycle

Understand releases and deployment

Identify project opportunities to apply CD

Explain things to consider when designing your release strategy

Define the components of a release pipeline and use artifact sources

Create a release approval plan

Implement release gates

Differentiate between a release and a deployment

Module 5: Implement a secure continuous deployment using Azure Pipelines

Lessons

Introduction to deployment patterns

Implement blue-green deployment and feature toggles

Implement canary releases and dark launching

Implement A/B testing and progressive exposure deployment

Integrate with identity management systems

Manage application configuration data

Lab : Configuring pipelines as code with YAML

Lab : Setting up and running functional tests

Lab : Integrating Azure Key Vault with Azure DevOps

After completing this module, students will be able to:

Explain the terminology used in Azure DevOps and other Release Management Tooling

Describe what a Build and Release task is, what it can do, and some available

deployment tasks

Implement release jobs

Differentiate between multi-agent and multi-configuration release job

Provision and configure target environment

Deploy to an environment securely using a service connection

Configure functional test automation and run availability tests

Setup test infrastructure

Use and manage task and variable groups

Module 6: Manage infrastructure as code using Azure and DSC

Lessons

Explore infrastructure as code and configuration management

Create Azure resources using Azure Resource Manager templates

Create Azure resources by using Azure CLI

Explore Azure Automation with DevOps

Implement Desired State Configuration (DSC)

Implement Bicep

Lab : Azure deployments using Azure Resource Manager templates

After completing this module, students will be able to:

Understand how to deploy your environment

Plan your environment configuration

Choose between imperative versus declarative configuration

Explain idempotent configuration

Create Azure resources using ARM templates

Understand ARM templates and template components

Manage dependencies and secrets in templates

Organize and modularize templates

Create Azure resources using Azure CLI

Module 7: Implement security and validate code bases for compliance

Lessons

Introduction to Secure DevOps

Implement open-source software

Software Composition Analysis

Static analyzers

OWASP and Dynamic Analyzers

Security Monitoring and Governance

Lab : Implement security and compliance in Azure Pipelines

Lab : Managing technical debt with SonarQube and Azure DevOps

After completing this module, students will be able to:

Identify SQL injection attack

Understand DevSecOps

Implement pipeline security

Understand threat modeling

Implement open-source software

Explain corporate concerns for open-source components

Describe open-source licenses

Understand the license implications and ratings

Work with Static and Dynamic Analyzers

Configure Microsoft Defender for Cloud

Module 8: Design and implement a dependency management strategy

Lessons

Explore package dependencies

Understand package management

Migrate, consolidate, and secure artifacts

Implement a versioning strategy

Introduction to GitHub Packages

Lab : Package management with Azure Artifacts

After completing this module, students will be able to:

Define dependency management strategy

Identify dependencies

Describe elements and componentization of a dependency management

Scan your codebase for dependencies

Implement package management

Manage package feed

Consume and create packages

Publish packages

Identify artifact repositories

Migrate and integrate artifact repositories

Module 9: Implement continuous feedback

Lessons

Implement tools to track usage and flow

Develop monitor and status dashboards

Share knowledge within teams

Design processes to automate application analytics

Manage alerts, Blameless retrospectives and a just culture

Lab : Monitoring application performance with Application Insights

Lab : Integration between Azure DevOps and Microsoft Teams

Lab : Sharing Team Knowledge using Azure Project Wikis

After completing this module, students will be able to:

Implement tools to track feedback

Plan for continuous monitoring

Implement Application Insights

Use Kusto Query Language (KQL)

Implement routing for mobile applications

Configure App Center Diagnostics

Configure alerts

Create a bug tracker

Configure Azure Dashboards

Work with View Designer in Azure Monitor

