

# Designing and Implementing Microsoft DevOps solutions

Course code: MOC AZ-400

This course provides the knowledge and skills to design and implement DevOps processes and practices. Students will learn how to plan for DevOps, use source control, scale Git for an enterprise, consolidate artifacts, design a dependency management strategy, manage secrets, implement continuous integration, implement a container build strategy, design a release strategy, set up a release management workflow, implement a deployment pattern, and optimize feedback mechanisms.

## At course completion students will be able

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and KPIs
- Create a team and agile organization structure
- Describe the benefits of using Source Control
- Migrate from TFVC to Git
- Scale Git for Enterprise DevOps
- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Migrate and consolidate artifacts
- Migrate and integrate source control measures
- Manage application config and secrets
- Develop a project quality strategy
- Plan for secure development practices and compliance rules
- Implement and manage build infrastructure
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps
- Manage code quality including: technical debt, SonarCloud, and other tooling solutions
- Manage security policies with open source, OWASP, and WhiteSource Bolt
- Implement a container strategy including how containers are different from virtual machines and how microservices use containers
- Implement containers using Docker
- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance
- Differentiate between a release and a deployment
- Define the components of a release pipeline

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Designing and Implementing Microsoft DevOps solutions

Explain things to consider when designing your release strategy

Classify a release versus a release process and outline how to control the quality of both

Describe the principle of release gates and how to deal with release notes and documentation

Explain deployment patterns, both in the traditional sense and in the modern sense

Choose a release management tool

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

Classify an Agent, Agent Queue, and Agent Pool

Explain why you sometimes need multiple release jobs in one release pipeline

Differentiate between multi-agent and multi-configuration release job

Use release variables and stage variables in your release pipeline

Deploy to an environment securely using a service connection

Embed testing in the pipeline

List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports

Create a release gate

Describe deployment patterns

Implement Blue Green Deployment

Implement Canary Release

Implement Progressive Exposure Deployment

Configure crash report integration for client applications

Develop monitoring and status dashboards

Implement routing for client application crash report data

Implement tools to track system usage, feature usage, and flow

Integrate and configure ticketing systems with development team's work management

Implement a mobile DevOps strategy

Apply infrastructure and configuration as code principles.

Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI

Describe deployment models and services that are available with Azure

Deploy and configure a Managed Kubernetes cluster

Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform

Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure

Implement compliance and security in your application infrastructure

Design practices to measure end-user satisfaction

## **GOPAS Praha**

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Brno**

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Bratislava**

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Designing and Implementing Microsoft DevOps solutions

Design processes to capture and analyze user feedback from external sources

Design routing for client application crash report data

Recommend monitoring tools and technologies

Recommend system and feature usage tracking tools

Analyze alerts to establish a baseline

Analyze telemetry to establish a baseline

Perform live site reviews and capture feedback for system outages

Perform ongoing tuning to reduce meaningless or non-actionable alerts

## Prerequisites

Knowledge in extent of the courses which are listed in the bellow sections **Previous Courses** and **Related Courses**

Good understanding of TCP/IP and DNS technologies

## Course outline

Transformation Planning

Project Selection

Team Structures

Migrating to Azure DevOps

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

Authenticating to Git in Azure Repos

How to Structure your Git Repo

Git Branching Workflows

Collaborating with Pull Requests in Azure Repos

Why care about GitHooks

Fostering Inner Source

Packaging Dependencies

Package Management

Migrating and Consolidating Artifacts

The concept of pipelines in DevOps

Azure Pipelines

Evaluate use of Hosted vs Private Agents

Agent Pools

Pipelines and Concurrency

Azure DevOps and Open Source Projects (Public Projects)

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Designing and Implementing Microsoft DevOps solutions

Azure Pipelines YAML vs Visual Designer  
Continuous Integration Overview  
Implementing a Build Strategy  
Integration with Azure Pipelines  
Integrate External Source Control with Azure Pipelines  
Set Up Private Agents  
Analyze and Integrate Docker Multi-Stage Builds  
Introduction to Security  
Implement secure and compliant development process  
Rethinking application config data  
Manage secrets, tokens, and certificates  
Implement tools for managing security and compliance in a pipeline  
Managing Code Quality  
Managing Security Policies  
Implementing a Container Build Strategy  
Package security  
Open source software  
Integrating license and vulnerability scans  
Implement a versioning strategy (git version)  
Introduction to Continuous Delivery  
Release strategy recommendations  
Building a High-Quality Release pipeline  
Choosing a deployment pattern  
Choosing the right release management tool  
Create a Release Pipeline  
Provision and Configure Environments  
Manage and Modularize Tasks and Templates  
Integrate Secrets with the release pipeline  
Configure Automated Integration and Functional Test Automation  
Automate Inspection of Health  
Introduction to Deployment Patterns  
Implement Blue Green Deployment  
Feature Toggles  
Canary Releases  
Dark Launching  
AB Testing

## **GOPAS Praha**

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Brno**

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Bratislava**

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Designing and Implementing Microsoft DevOps solutions

Progressive Exposure Deployment  
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  
Introduction to Mobile DevOps  
Introduction to Visual Studio App Center  
Manage mobile target device sets and distribution groups  
Manage target UI test device sets  
Provision tester devices for deployment  
Create public and private distribution groups  
Infrastructure as Code and Configuration Management  
Create Azure Resources using ARM Templates  
Create Azure Resources using Azure CLI  
Create Azure Resources by using Azure PowerShell  
Desired State Configuration (DSC)  
Azure Automation with DevOps  
Additional Automation Tools  
Deployment Modules and Options  
Azure Infrastructure-as-a-Service (IaaS) Services  
Azure Platform-as-a-Service (PaaS) services  
Serverless and HPC Computer Services  
Azure Service Fabric  
Azure Kubernetes Service  
Chef  
Puppet  
Ansible  
Terraform  
Security and Compliance Principles with DevOps  
Azure security Center  
The inner loop  
Continuous Experimentation mindset  
Design practices to measure end-user satisfaction  
Design processes to capture and analyze user feedback  
Design process to automate application analytics  
Site Reliability Engineering

## **GOPAS Praha**

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Brno**

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

## **GOPAS Bratislava**

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Designing and Implementing Microsoft DevOps solutions

Analyze telemetry to establish a baseline

Perform ongoing tuning to reduce meaningless or non-actionable alerts

Analyze alerts to establish a baseline

Blameless Retrospectives and a Just Culture

## Preparation for Microsoft certification

Most Microsoft certification exams do not require students to attend an official MOC course in order to pass the exam.

This applies to all certifications except for MCM

Official Microsoft MOC courses as well as our own GOC courses are good ways of preparation for Microsoft certifications such as MCP, MTA, MCSA, MCSE or MCM

This does not mean that official MOC courses would serve as the only necessary preparation. The primary goal of an MOC course is to provide for sufficient theoretical knowledge and practical experience to effectively work with the related product

MOC courses usually cover most of the topics required by their respective certification exams, but often do not give every topic the same amount of time and emphasis as may be required to completely pass the exam.

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved