

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

**Course Number:** MOC-AZ-400  
**Duration:** 4 days

**Overview**

This Designing and Implementing Microsoft DevOps solutions training (Microsoft course AZ-400) teaches attendees how to design and implement DevOps processes and practices for the Azure platform. Attendees learn how to integrate Azure with pipelines, infrastructure and configuration tools, automation, deployment, and more. This course prepares students for the [AZ-400 exam](https://docs.microsoft.com/en-us/learn/certifications/exams/AZ-400) for which every attendee receives a voucher.

**Prerequisites**

* Cloud computing concepts, including an understanding of PaaS, SaaS, and IaaS implementations
* Both Azure administration and Azure development with proven expertise in at least one of these areas
* Version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software

**Materials**

All Microsoft training students receive Microsoft official courseware.

For all Microsoft Official Courses taught in their entirety that have a corresponding certification exam, an exam voucher is included for each participant.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

* Design and implement DevOps processes and practices
* 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
* Optimize feedback mechanisms

**Outline**

* 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
* Structure your Git Repo
* Manage Git branches and workflows
* Collaborate with pull requests in Azure Repos
* Identify technical debt
* Explore Git hooks
* Plan foster inner source
* Manage Git repositories
* 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
* Introduction to continuous delivery
* Create a release pipeline
* Explore release recommendations
* Provision and test environments
* Manage and modularize tasks and templates
* Automate inspection of health
* 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
* 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
* Introduction to Secure DevOps
* Implement open-source software
* Software Composition Analysis
* Static analyzers
* OWASP and Dynamic Analyzers
* Security Monitoring and Governance
* Explore package dependencies
* Understand package management
* Migrate consolidating and secure artifacts
* Implement a versioning strategy
* Introduction to GitHub Packages
* 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