

Microsoft AZ-400 Microsoft Azure DevOps Solutions

Microsoft Official Course

AZ-400 Microsoft Azure DevOps Solutions
for

Microsoft Certified: Azure DevOps Engineer Expert



About this Course

This course provides the knowledge and skills to implement DevOps processes. Students will learn how to use source control, scale Git for an enterprise, and implement and manage build infrastructure. We will also look at continuous integration, learning how to implement continuous integration in an Azure DevOps pipeline, managing code quality and security principles and implementing a container build strategy.

The course focuses on implementing continuous delivery with students learning how to design a release strategy, set up a release management workflow, and implement an appropriate

deployment pattern, and implementing dependency management where students learn how to design a dependency management strategy and manage security and compliance.

We examine the deployment of application infrastructure in DevOps pipelines – how to implement infrastructure as code and configuration management, how to provision Azure infrastructure using common automation tools, and how to deploy an application infrastructure using various Azure services and deployment methodologies. Students will also learn how to integrate 3rd party deployment tools with Azure, such as Chef and Puppet to incorporate compliance and security into the release pipeline.

Our course provides the knowledge and skills needed to implement continuous feedback, working on how to recommend and design system feedback mechanisms, implement a process for routing system feedback to development teams, and optimize feedback mechanisms.

Finally we will look at designing a DevOps strategy. Students will learn how to plan for transformation, select a project, and create team structures. They will also learn how to develop quality and security strategies, and planning for migrating and consolidating artifacts and source control will also be covered.

Course Goals/ At the end of the Course you will be able to:

- Describe the benefits of using source control
- Migrate from TFVC to Git
- Scale Git for Enterprise DevOps
- Implement and manage build infrastructure
- Manage application config & secrets

- Implement a mobile DevOps strategy
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps
- Configure builds and the options available
- Create an automated build workflow
- Integrate other build tooling with Azure DevOps
- Create hybrid build processes
- Describe what is meant by code quality and how it is measured
- Detect code smells
- Integrate automated tests for code quality
- Report on code coverage during testing
- Add tooling to measure technical debt
- Detect open source and other licensing issues
- Implement a container build strategy
- Differentiate between a release and a deployment
- Define the components of a release pipeline
- 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
- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- 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
- 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
- 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
- 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
- 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 system
- 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
- Plan for the transformation with shared goals and timelines.
- Select a project and identify project metrics and KPIs.
- Create a team and agile organizational structure.
- Develop a project quality strategy.
- Plan for secure development practices and compliance rules.
- Migrate and consolidate artifacts.
- Migrate and integrate source control measures.

Targeted Audience

- Students in this course are interested in implementing DevOps processes like dependency management, application infrastructure and the management and configuration of that infrastructure; or in passing the Microsoft Azure DevOps Solutions certification exam.

Course Format



Classroom Onsite



Virtual anywhere

Language: Bulgarian or English

Course Duration:

- 5 work days (monday – friday 09:00 – 17:00)

or


- **40 hours out of hours learning**

Payments:




Next Classes:

Jun 6

6 June 2026 @ 10:00 - 28 June 2026 @ 14:00 EEST 

[Linux Professional Institute LPIC-1 prep for 101-500 and 102-500 \(4 weeks\)](#)

Jun 22

22 June 2026 @ 09:00 - 26 June 2026 @ 17:00 EEST 

[EC-Council – Certified Ethical Hacker \(CEHv13 EN\)](#)

Jun 29

29 June 2026 @ 09:00 - 3 July 2026 @ 17:00 EEST 

[EC-Council – Certified Ethical Hacker \(CEHv13 EN\)](#)

Jul 4

4 July 2026 @ 10:00 - 26 July 2026 @ 14:00 EEST 

Linux Professional Institute LPIC-1 prep for 101-500 and 102-500 (4 weeks)

Jul 6

6 July 2026 @ 09:00 - 10 July 2026 @ 17:00 EEST 📅

EC-Council – Certified Ethical Hacker (CEHv13 EN)

[View Calendar](#)

For more information, use the contact format. We will contact you to confirm the data.

Prerequisites:

- Fundamental knowledge about Azure, version control, Agile software development, and core software development principles is required, along with experience with virtual machines and containers, and some exposure to automation and scripting. Students should also have knowledge of general application development and deployment processes. It is recommended that you have experience working in an IDE, as well as some knowledge of the Azure portal. However, students who may not have a technical background in these technologies, but who are curious about DevOps practices as a culture shift, should be able to follow the procedural and expository explanations of continuous integration regardless. Experience working in a software development or operations environment with either Windows or Linux would be

helpful but is not essential, as would experience in an organization that delivers software. Students who attend this course are interested in implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam.

This Class will teach you how to pass the following exams:

- AZ-400 Microsoft Azure DevOps Solutions and achieve the Microsoft Certified: Azure DevOps Engineer Expert Certification