

Designing and Implementing Microsoft DevOps solutions (AZ-400T00)

Duration: 4 days

Audience: Professionnel TI

Who is this training for?

Attendees want to implement DevOps processes or pass the Microsoft Azure DevOps Solutions certification exam.

Training objectives

- Plan the transformation with shared objectives and deadlines.
- Select a project and identify project metrics and key performance indicators (KPIs).
- Create an agile team and organizational structure. • Design a tool integration strategy.
- Design a licensing strategy (e.g., Azure DevOps and GitHub users).
- Design an end-to-end traceability strategy from work items to functional software.
- Design an authentication and access strategy.
- Design a strategy for integrating on-premises and cloud resources.
- Describe the benefits of using source control.
- Describe Azure Repos and GitHub.
- Migrate from TFVC to Git.
- Manage code quality, including SonarCloud technical debt and other tooling solutions.
- Build organizational knowledge on code quality.
- Explain how to structure Git repositories.
- Describe Git branching workflows.
- Leverage pull requests for collaboration and code reviews.
- Leverage Git hooks for automation.
- Use Git to foster open source throughout the organization.
- Explain the role of Azure pipelines and its components.
- Configure agents for use in Azure pipelines.

- Explain why continuous integration is important.
- Implement continuous integration using Azure pipelines.
- Define Site Reliability Engineering (SRE).
- Design processes to measure end-user satisfaction and analyze user feedback.
- Design processes to automate application analysis.
- Manage alerts and reduce non-significant and non-actionable alerts.
- Perform flawless retrospectives and create a fair culture.
- Define an infrastructure and configuration strategy and an appropriate set of tools for a release pipeline.

Summary

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

Course outline

Learning Path 1: Implement Development for Enterprise DevOps.

- Introduction to DevOps.
- Plan for Agile with GitHub Projects and Azure Boards.
- Design and implement branch strategies and workflows.
- Collaborate with pull requests in Azure Repos.
- Explore Git hooks.
- Favor the inner source.
- Manage and configure repositories.
- Identify technical debt.

Learning path 2: Implement continuous integration with Azure Pipelines and GitHub Actions.

- Explore Azure Pipelines.
- Manage Azure Pipelines agents and pools.
- Describe pipelines and competition.
- Design and implement a pipeline strategy.
- Integrate with Azure Pipelines.
- Introduction to GitHub Actions.
- Learn continuous integration with GitHub Actions.
- Design a container creation strategy.

Learning Path 3: Design and implement a release strategy.

- Create a release pipeline.
- Explore publication recommendations.
- Provide and test environments.
- Manage and modularize tasks and templates.
- Automate condition inspection.

Learning path 4: Implement secure continuous deployment with Azure Pipelines.

- Introduction to deployment models.
- Implement blue-green deployment and feature switches.
- Implement canary deployments and dark launches.
- Implement A/B testing and phased-in deployments.
- Integrate with identity management systems.
- Manage application configuration data.

Learning path 5: Manage infrastructure as code with Azure and DSC.

- Explore infrastructure as code and configuration management.
- Create Azure resources using Azure Resource Manager templates.
-

Create Azure resources with the Azure CLI. Explore Azure Automation with DevOps.

- Implement Desired State Configuration (DSC). Implement Bicep.

Learning Path 6: Implement security and validate codebases for compliance.

- Introduction to Secure DevOps.
- Implement open-source software.
- Software composition analysis.
- Security oversight and governance.
- Learning

Path 7: Design and implement a dependency management strategy.

- Explore package dependencies.
- Understand package management.
- Migrate, consolidate, and secure artifacts.
- Implement a versioning strategy.
- Introduction to GitHub Packages.

Learning Path 8: Implement Continuous Feedback.

- Implement tools to track usage and flow.
- Develop, monitor, and create status dashboards.
- Share knowledge within teams.
- Design processes to automate application scanning.
- Managing alerts, blame-free retrospectives, and a fair culture.

Approach and methodology

The course takes a pedagogical approach that combines structured theoretical presentations on DevOps principles, real-world scenario analysis in an organizational setting, and guided hands-on work using Azure DevOps and GitHub tools. Participants conduct design workshops on CI/CD

pipelines, deployment strategies, and DevSecOps practices. The entire approach is directly aligned with the goals of the Microsoft Certification exam.

Recommendations

- Cloud computing concepts, including an understanding of PaaS, SaaS, and IaaS implementations.
- Azure administration and Azure development with proven know-how in at least one of these areas.
- Version control, agile software development, and basic software development principles. It would be beneficial to have the experience of an organization that provides software.

Certifications

- Exam: AZ-400 – Designing and Implementing Microsoft DevOps Solutions
- Certification obtained: Microsoft Certified: DevOps Engineer Expert
- Link to training: Training prepares directly for exam AZ-400