Designing and Implementing Microsoft DevOps solutions (AZ-400)

Description

A Comprehensive Training to Master DevOps with AZ-400

Are you looking to enhance your DevOps skills and earn the AZ-400 certification? Our training is specifically designed for IT professionals aiming to optimize their DevOps practices and processes. With a hands-on approach focused on real-world scenarios, you'll learn continuous integration, manage complex pipelines with Azure Pipelines, and automate infrastructure management using Azure and tools like Terraform and Ansible.

Master DevOps Tools and Methods

Whether you are a developer or a system administrator, this course will equip you with the skills to orchestrate end-to-end DevOps workflows. From getting started with Git to advanced container management with Docker and Kubernetes, we will guide you in creating an agile, secure, and high-performing environment.

Prepare for Your Future with the AZ-400 Certification

Our training program not only prepares you to succeed in the AZ-400 certification but also equips you with operational skills that are immediately applicable in the workplace. Join us to turn your DevOps projects into success stories!

Course Content
Module 1: Introduction to DevOps

- What is DevOps?
- Explore the DevOps journey
- Identify transformation teams
- Define the organization structure for agile practices
- Explore shared goals and define timelines
- What is Azure DevOps?
- What is GitHub?
- Design a license management strategy
- What is source control?

- Describe local Git usage
- Introduction to Azure Repos
- Introduction to GitHub

Module 2: Agile planning with GitHub Projects and Azure Boards

- Overview of GitHub Projects and project boards
- Overview of Azure Boards
- Configure projects and teams in Azure DevOps
- Link GitHub to Azure Boards
- Set up GitHub projects
- Manage work using GitHub project boards
- Customize project views
- Collaborate using team discussions
- Design and implement a strategy for feedback loops
- Design and implement traceability from source to bugs and quality
- Agile planning and portfolio management with Azure Boards

Module 3: Design and implement branching strategies and workflows

- Explore types of branching workflows
- Explore a feature branch workflow
- Explore the Git branching model for continuous delivery
- Explore GitHub flow
- Explore the fork workflow
- Implement branch merge restrictions
- Version control with Git in Azure Repos

Module 4: Collaborate with pull requests in Azure Repos

- Collaborate using pull requests

Module 5: Explore Git hooks

- Overview of Git hooks
- Implement Git hooks

Module 6: Plan inner source promotion

- Explore inner source promotion
- Implement the fork workflow
- Describe inner source using forks

Module 7: Manage and configure repositories

- Work with large repositories
- Explore monorepo vs multi-repo
- Implement a changelog
- Use Scalar and share cross-repos
- Retrieve specific data using Git commands
- Purge repository data
- Manage versions with GitHub Repos
- Automate release notes with GitHub

- Create API documentation
- Automate Git history documentation
- Set repository permissions using GitHub
- Configure GitHub tags to organize repositories

Module 8: Identify technical debt

- Review code quality
- Review complexity and quality metrics
- Introduction to technical debt
- Measure and manage technical debt
- Introduction to GitHub Advanced Security
- Integrate other code quality tools
- Plan effective code reviews

Module 9: Explore Azure Pipelines

- Understand the concept of pipelines in DevOps
- Describe Azure Pipelines
- Understand key Azure Pipelines terms

Module 10: Manage Azure Pipelines pools and agents

- Choose between Microsoft-hosted or self-hosted agents
- Explore job types
- Overview of agent pools
- Explore a predefined agent pool
- Understand typical agent pool scenarios
- Communicate with Azure Pipelines
- Communicate for deployment to target servers
- Review other considerations
- Describe agent pool security
- Configure agent pools and understand pipeline styles

Module 11: Describe pipelines and concurrency

- Understand parallel jobs
- Estimate parallel jobs
- Describe Azure Pipelines and open source projects
- Explore Azure Pipelines and the Visual Designer
- Describe Azure Pipelines and YAML
- Enable continuous integration with Azure Pipelines

Module 12: Design and implement a pipeline strategy

- Configure agent demands
- Explore multiple configurations and multiple agents
- Integrate GitHub repositories with Azure Pipelines
- Design and implement a complete test strategy
- Implement code coverage and display it in the pipeline
- Implement multi-job builds
- Explore supported source control types in Azure Pipelines

Module 13: Integrate with Azure Pipelines

- Describe the anatomy of a pipeline
- Understand the structure of a pipeline
- Detail templates
- Explore YAML resources
- Use multiple repositories in your pipeline
- Migrate a classic pipeline to YAML in Azure Pipelines

Module 14: Introduction to GitHub Actions

- What are GitHub Actions?
- Explore the actions flow
- Understand workflows
- Describe standard workflow syntax elements
- Explore events
- Explore jobs
- Explore runners
- Review and test an action

Module 15: Discover continuous integration with GitHub Actions

- Describe CI with Actions
- Review environment variables
- Share artifacts between jobs
- Review workflow badges
- Describe best practices for creating Actions
- Tag releases with Git labels
- Create encrypted secrets
- Use secrets in a workflow
- Implement GitHub Actions for CI/CD

Module 16: Design a container build strategy

- Review container structure
- Use Docker containers
- Understand Dockerfile fundamentals
- Review multi-stage Dockerfiles
- Review multi-stage build considerations
- Explore Azure container-related services
- Deploy Docker containers to Azure App Service

Module 17: Create a release pipeline

- Describe Azure DevOps release pipeline features
- Explore release pipelines
- Explore artifact sources
- Select the appropriate artifact source
- Review phased deployment considerations
- Explore build and release tasks
- Explore custom build and release tasks
- Explore release jobs
- Understand database deployment tasks

- Configure pipelines as code with YAML

Module 18: Explore release recommendations

- Understand delivery cadence and trigger types
- Explore release approvals
- Explore release gates
- Use gates to ensure quality
- Explore GitOps recommendations and strategies
- Control deployments with Release Gates

Module 19: Provision and test environments

- Provision and configure target environments
- Configure automation for integration and functional testing
- Overview of shift-left approach
- Configure and run availability tests
- Explore Azure Load Testing
- Configure and run functional tests

Module 20: Manage and modularize tasks and templates

- Review task groups
- Explore deployment pipeline variables

Module 21: Automate health inspection

- Automate health checks
- Explore events and notifications
- Explore service hooks
- Configure Azure DevOps notifications
- Configure GitHub notifications
- Review how to measure release process quality
- Review release notes and documentation
- Review criteria for choosing release management tools
- Explore common release management tools

Module 22: Introduction to deployment models

- Explore microservices architecture
- Review classical deployment models
- Understand modern deployment models

Module 23: Implement blue-green deployment and feature toggles

- What is blue-green deployment?
- Explore deployment slots
- Overview of feature toggles
- Describe feature toggle maintenance

Module 24: Implement releases with approval and dark launching

- Explore releases with approval checks

- Review Traffic Manager
- Understand dark launching

Module 25: Implement A/B testing and progressive exposure deployment

- What is A/B testing?
- Explore CI-CD with deployment rings

Module 26: Integrate with identity administration systems

- Integrate GitHub with single sign-on (SSO)
- Design and implement roles and permissions in GitHub
- Design and implement permissions and security groups in Azure DevOps
- Explore workload identities
- Implement managed identities

Module 27: Manage application configuration data

- Rethink application configuration data
- Explore separation of concerns
- Understand external configuration store models
- Implement secure files in Azure DevOps
- Overview of Azure App Configuration
- Review key-value pairs
- Review application feature configuration management
- Integrate Azure Key Vault with Azure Pipelines
- Manage secrets, tokens, and certificates
- Review DevOps inner and outer loop
- Integrate Azure Key Vault with Azure DevOps
- Enable dynamic configuration and feature flags

Module 28: Explore infrastructure as code and configuration management

- Explore environment deployment
- Review environment configuration
- Understand imperative and declarative configuration
- Overview of idempotent configuration

Module 29: Create Azure resources using ARM templates

- Why use ARM templates?
- Explore template components
- Manage dependencies
- Modularize templates
- Manage secrets in templates

Module 30: Create Azure resources using Azure CLI

- What is Azure CLI?
- Use Azure CLI

Module 31: Explore Azure Automation with DevOps

- Create automation accounts
- What is a runbook?
- Understand shared automation resources
- Explore runbook gallery
- Review webhooks
- Explore source control integration
- Explore PowerShell workflows
- Create a workflow
- Explore hybrid management
- Review checkpoint and parallel processing

Module 32: Implement Desired State Configuration (DSC)

- Understand configuration drift
- Explore Desired State Configuration (DSC)
- Explore Azure Automation State Configuration (DSC)
- Review DSC configuration file
- Implement Linux automation and DSC on Azure

Module 33: Implement Bicep

- What is Bicep?
- Install Bicep
- Understand Bicep file structure and syntax
- Deploy a Bicep file from Azure Pipelines
- Deploy a Bicep file from GitHub workflows
- Deploy using Azure Bicep templates

Module 34: Introduction to Secure DevOps

- Describe SQL injection attacks
- Understand DevSecOps
- Explore secure DevOps pipeline
- Explore key validation points
- Explore continuous security validation
- Understand threat modeling
- Explore CodeQL in GitHub

Module 35: Implement open source software

- Learn how software is built
- What is open source software?
- Explore business concerns with open source components
- Overview of open source licenses
- Explore common open source licenses
- Review license implications and assessments

Module 36: Software composition analysis

- Inspect and validate codebase compliance
- Explore software composition analysis (SCA)
- Implement GitHub Dependabot alerts and security updates

- Integrate SCA checks into pipelines
- Review tools to assess package security and licensing
- Automate container image analysis
- Interpret scanner tool alerts

Module 37: Security monitoring and governance

- Implement pipeline security
- Explore Microsoft Defender for Cloud
- Review Defender for Cloud use cases
- Review Azure Policy
- Understand policies
- Explore initiatives
- Explore resource locks
- Understand Microsoft Defender for Identity
- Integrate GitHub Advanced Security with Defender for Cloud
- Configure GitHub Advanced Security for GitHub and Azure DevOps

Module 38: Explore package dependencies

- What is dependency management?
- Describe elements of a dependency management strategy
- Identify dependencies
- Understand componentization via source or packages
- Decompose your system
- Analyze your codebase for dependencies

Module 39: Understand package management

- Explore packages
- Understand package feeds
- Explore package feed managers
- Explore common public package sources
- Explore self-hosted and SaaS package sources
- Consume packages
- Overview of Azure Artifacts
- Publish packages
- Manage packages with Azure Artifacts

Module 40: Migrate, consolidate, and secure artifacts

- Identify existing artifact repositories
- Migrate and integrate artifact repositories
- Secure access to package feeds
- Review roles
- Review permissions
- Review authentication

Module 41: Implement a versioning strategy

- Understand artifact version control
- Explore semantic versioning
- Review final version views

- Promote packages
- Explore best practices for version control

Module 42: Introduction to GitHub Packages

- Publish packages
- Install a package
- Delete and restore a package
- Explore package access control and visibility

Module 43: Implement tools to track usage and flow

- Understand the inner loop
- Overview of continuous monitoring
- Explore Azure Monitor and Log Analytics
- Review Kusto Query Language (KQL)
- Explore Application Insights
- Implement Application Insights
- Design and implement metrics and queries
- Monitor application performance with Azure Load Testing

Module 44: Develop monitoring and status dashboards

- Configure monitoring in GitHub
- Explore Azure dashboards
- Explore Azure Monitor workbooks
- Explore Power BI
- Create your own custom application
- Monitor pipeline health including failure rate, duration, and flaky tests
- Optimize a pipeline for cost, time, performance, and reliability
- Optimize pipeline concurrency for performance and cost

Module 45: Share knowledge within teams

- Share learnings within development teams
- Overview of Azure DevOps project wikis
- Integrate Microsoft Teams with Azure DevOps and GitHub
- Share team knowledge using Azure Project Wiki

Module 46: Design processes to automate application analysis

- Explore rapid response and augmented search
- Integrate telemetry
- Review monitoring tools and technologies
- Explore IT service management connector

Module 47: Manage alerts, blameless retrospectives, and a just culture

- Review when to receive a notification
- Learn how to resolve issues
- Explore smart detection notifications
- Improve performance
- Understand server response time degradation
- Reduce meaningless and unactionable alerts

- Analyze blameless retrospectives
- Foster a just culture

Lab / Exercises

- This course provides you with exclusive access to the official Microsoft lab, enabling you to practice your skills in a professional environment.

Documentation

- Access to Microsoft Learn, Microsoft's online learning platform, offering interactive resources and educational content to deepen your knowledge and develop your technical skills.

Exam

- This course prepares you to the AZ-400: Designing and Implementing Microsoft DevOps Solutions exam.

Participant profiles

- People interested in designing and implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam

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

Objectives

- Plan for the transformation with shared goals and timelines
- 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

Description
Designing and Implementing Microsoft DevOps solutions (AZ-400)
Niveau
Avancé
Classroom Registration Price (CHF)
3200
Virtual Classroom Registration Price (CHF)
3000
Duration (in Days)
4
Reference
AZ-400T00