

Designing and Implementing Microsoft DevOps solutions

Code:	AZ-400T00
Length:	5 days
URL:	View Online

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.

Skills Gained

- 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
- 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
- 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

Who Can Benefit

Students in this course are interested in designing and implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam.

Prerequisites

Successful learners will have prior knowledge and understanding of:

- 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.

If you are new to Azure and cloud computing, consider one of the following resources:

- Free online: Azure Fundamentals (<https://docs.microsoft.com/en-us/learn/paths/azure-fundamentals/>)
- Instructor-led course: AZ-900: Azure Fundamentals (<https://docs.microsoft.com/en-us/learn/certifications/courses/az-900t01>)

If you are new to Azure Administration, consider taking:

- Free online: Prerequisites for Azure Administrators (<https://docs.microsoft.com/en-us/learn/paths/az-104-administrator-prerequisites/>)
- Instructor-led courses: AZ-104: Microsoft Azure Administrator (<https://docs.microsoft.com/en-us/learn/certifications/courses/az-104t00>) and AZ-010: Azure Administration for AWS SysOps (<https://docs.microsoft.com/en-us/learn/certifications/courses/az-010t00>)

If you are new to Azure Developer, consider taking:

- Free online: Create serverless applications (<https://docs.microsoft.com/en-us/learn/paths/create-serverless-applications/>)
- Instructor-led course: AZ-204: Developing Solutions for Microsoft Azure (<https://docs.microsoft.com/en-us/learn/certifications/courses/az-204t00>) and AZ-020: Microsoft Azure Solutions for AWS Developers (<https://docs.microsoft.com/en-us/learn/certifications/courses/az-020t00>)

Related Certifications

Authorized training from ExitCertified is created and maintained by the vendor who also creates the certification exams. While it may not be a requirement of certification to attend a vendor-authorized training class, doing so will put you in the best position to successfully complete the related exams. Start training and begin working towards one of the following certifications today.

MICROSOFT AZURE APPS AND INFRASTRUCTURE

Microsoft Azure DevOps Engineer Expert

[View Certification arrow_forward](#)

[View All Certifications arrow_forward](#)

Refer a friend or colleague and get up to \$100 Amazon gift card* — when they book training!

[Learn More](#)

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.

Generated 12