Course Outline

AZ-400T00 A: Designing and Implementing Microsoft DevOps Solutions

Course Summary

Description

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.

Objectives

At the end of this course, students will be able to:

- 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

Course Summary (cont'd)

Topics

- 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

Audience

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

Prerequisites

Fundamental knowledge about Azure, version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

Duration

Four days

Course Outline

Course Outline

I. Introduction to DevOps
- A. Understand what DevOps is and the steps to accomplish it
- B. Identify teams to implement the process
- C. Plan for the transformation with shared goals and timelines
- D. Plan and define timelines for goals

II. Choose the right project
- A. Understand different projects and systems to guide the journey
- B. Select a project to start the DevOps transformation
- C. Identify groups to minimize initial resistance
- D. Identify project metrics and Key Performance Indicators (KPI's)

III. Describe team structures
- A. Understand agile practices and principles of agile development
- B. Create a team and agile organizational structure
- C. Identify ideal DevOps team members
- D. Select and configure tools for collaboration

IV. Choose the DevOps tools
- A. Design a tool integration strategy
- B. Design a license management strategy (e.g. Azure DevOps and GitHub users)
- C. Design a strategy for end-to-end traceability from work items to working software
- D. Design an authentication and access strategy
- E. Design a strategy for integrating on-premises and cloud resources

V. Plan Agile with GitHub Projects and Azure Boards
- A. Describe GitHub Projects and Azure Boards
- B. Link Azure Boards and GitHub
- C. Configure and Manage GitHub Projects and boards
- D. Customize Project views

VI. Introduction to source control
- A. Understand source control
- B. Apply best practices for source control
- C. Describe the benefits of using source control

VII. Describe types of source control systems
- A. Apply source control practices in your development process
- B. Explain differences between centralized and distributed version control
- C. Understand Git and TFVC
- D. Develop using Git

VIII. Work with Azure Repos and GitHub
- A. Describe Azure Repos and GitHub
- B. Migrate from TFVC to Git
- C. Work with GitHub Codespaces

IX. Structure your Git Repo
- A. Understand Git repositories
- B. Implement mono repo or multiple repos
- C. Explain how to structure Git Repos
- D. Implement a change log

X. Manage Git branches and workflows
- A. Describe Git branching workflows
- B. Implement feature branches
- C. Implement GitHub Flow
- D. Fork a repo

XI. Collaborate with pull requests in Azure Repos
- A. Leverages pull requests for collaboration and code reviews
- B. Give feedback using pull requests
- C. Configure branch policies
- D. Use GitHub mobile for pull requests approvals

XII. Identify technical debt
- A. Identify and manage technical debt
- B. Integrate code quality tools
- C. Plan code reviews
- D. Describe complexity and quality metrics

XIII. Explore Git hooks
- A. Understand Git hooks
- B. Identify when used Git hooks
- C. Implement Git hooks for automation
- D. Explain Git hooks' behavior

Course Outline (cont'd)

XIV. Plan foster inner source
- A. Use Git to foster inner source across the organization
- B. Implement fork workflow
- C. Choose between branches and forks
- D. Share code between forks

XV. : Manage Git repositories
- A. Understand large Git repositories
- B. Explain VFS for Git
- C. Use Git Large File Storage (LFS)
- D. Purge repository data
- E. Manage and Automate Release Notes with GitHub

XVI. Explore Azure Pipelines
- A. Describe Azure Pipelines
- B. Explain the role of Azure Pipelines and its components
- C. Decide Pipeline automation responsibility
- D. Understand Azure Pipeline key terms

XVII. Manage Azure Pipeline agents and pools
- A. Choose between Microsoft-hosted and self-hosted agents
- B. Install and configure Azure Pipelines Agents
- C. Configure agent pools
- D. Make the agents and pools secure
- E. Explore communication to deploy using Azure Pipelines

XVIII. Describe pipelines and concurrency
- A. Use and estimate parallel jobs
- B. Use Azure Pipelines for open-source or private projects
- C. Use Visual Designer
- D. Work with Azure Pipelines and YAML

XIX. Explore continuous integration
- A. Explain why continuous integration matters
- B. Implement continuous integration using Azure Pipelines
- C. Explain benefits of continuous integration
- D. Describe build properties

XX. Implement a pipeline strategy
- A. Define a build strategy
- B. Explain and configure demands
- C. Implement multi-agent builds
- D. Use different source control types available in Azure Pipelines

XXI. Integrate with Azure Pipelines
- A. Describe advanced Azure Pipelines anatomy and structure
- B. Detail templates and YAML resources
- C. Implement and use multiple repositories

XXII. Introduction to GitHub Actions
- A. Explain GitHub Actions and workflows
- B. Create and work with GitHub Actions and Workflows
- C. Describe Events, Jobs and Runners
- D. Examine output and release management for actions

XXIII. Learn continuous integration with GitHub Actions
- A. Implement Continuous Integration with GitHub Actions
- B. Use environment variables
- C. Share artifacts between jobs and use Git tags
- D. Create and manage secrets

XXIV. Design a container build strategy
- A. Design a container strategy
- B. Work with Docker Containers
- C. Create an Azure Container Registry
- D. Explain Docker microservices and containers

XXV. Introduction to continuous delivery
- A. Explain continuous delivery (CD)
- B. Implement continuous delivery in your development cycle
- C. Understand releases and deployment
- D. Identify project opportunities to apply CD

Course Outline (cont'd)

XXVI. Create a release pipeline
- A. Explain the terminology used in Azure DevOps and other Release Management Tooling
- B. Describe what a Build and Release task is, what it can do, and some available deployment tasks
- C. Implement release jobs

XXVII. Explore release recommendations
- A. Explain things to consider when designing your release strategy
- B. Define the components of a release pipeline and use artifact sources
- C. Create a release approval plan
- D. Implement release gates

XXVIII. Provision and test environments
- A. Provision and configure target environment
- B. Deploy to an environment securely using a service connection
- C. Configure functional test automation and run availability tests
- D. Setup test infrastructure

XXIX. Manage and modularize tasks and templates
- A. Use and manage task and variable groups
- B. Use release variables and stage variables in your release pipeline
- C. Use variables in release pipelines

XXX. Automate inspection of health
- A. Implement automated inspection of health
- B. Create and configure events
- C. Configure notifications in Azure DevOps and GitHub
- D. Create service hooks to monitor pipeline
- E. Classify a release versus a release process, and outline how to control the quality of both
- F. Choose a release management tool

XXXI. Introduction to deployment patterns
- A. Describe deployment patterns
- B. Explain microservices architecture
- C. Understand classical and modern deployment patterns
- D. Plan and design your architecture

XXXII. Implement blue-green deployment and feature toggles
- A. Explain deployment strategies
- B. Implement blue green deployment
- C. Understand deployment slots
- D. Implement and manage feature toggles"

XXXIII. Implement canary releases and dark launching
- A. Describe deployment strategies
- B. Implement canary release
- C. Explain traffic manager
- D. Understand dark launching

XXXIV. Implement A/B testing and progressive exposure deployment
- A. Implement progressive exposure deployment
- B. Implement A/B testing
- C. Implement CI/CD with deployment rings
- D. Identify the best deployment strategy

XXXV. Integrate with identity management systems
- A. Integrate Azure DevOps with identity management systems
- B. Integrate GitHub with single sign-on (SSO)
- C. Understand and create a service principal
- D. Create managed service identities

XXXVI. Manage application configuration data
- A. Rethink application configuration data
- B. Understand separation of concerns
- C. Integrate Azure Key Vault with Azure Pipelines
- D. Manage secrets, tokens and certificates
- E. Describe Azure App Configuration
- F. Understand Key-value pairs
- G. Understand app configuration feature management

Course Outline (cont'd)

XXXVII. Explore infrastructure as code and configuration management

A. Understand how to deploy your environment
B. Plan your environment configuration
C. Choose between imperative versus declarative configuration
D. Explain idempotent configuration

XXXVIII. Create Azure resources using Azure Resource Manager templates

A. Create Azure resources using Azure Resource Manager templates
B. Understand Azure Resource Manager templates and template components
C. Manage dependencies and secrets in templates
D. Organize and modularize templates

XXXIX. Create Azure resources by using Azure CLI

A. Create Azure resources using Azure CLI
B. Understand and work with Azure CLI
C. Run templates using Azure CLI
D. Explains Azure CLI commands

XL. Explore Azure Automation with DevOps

A. Implement automation with Azure DevOps
B. Create and manage runbooks
C. Create webhooks
D. Create and run a workflow runbook and PowerShell workflows

XLI. Implement Desired State Configuration (DSC)

A. Implement Desired State Configuration (DSC)
B. Describe Azure Automation State Configuration
C. Implement DSC and Linux Automation on Azure
D. Plan for hybrid management

XLII. Implement Bicep

A. Learn what Bicep is
B. Learn how to install it and create a smooth authoring experience
C. Use Bicep to deploy resources to Azure
D. Deploy Bicep files in Cloud Shell and Visual Studio Code

XLIII. Introduction to Secure DevOps

A. Identify SQL injection attack
B. Understand DevSecOps
C. Implement pipeline security
D. Understand threat modeling

XLIV. Implement open-source software

A. Implement open-source software
B. Explain corporate concerns for open-source components
C. Describe open-source licenses
D. Understand the license implications and ratings

XLV. Software Composition Analysis

A. Inspect and validate code bases for compliance
B. Integrate security tools like WhiteSource with Azure DevOps
C. Implement pipeline security validation
D. Interpret alerts from scanning tools
E. Configure GitHub Dependabot alerts and security

XLVI. Static analyzers

A. Understand Static Analyzers
B. Work with SonarCloud
C. Work with CodeQL in GitHub
D. Interpret alerts from scanning tools

XLVII. OWASP and Dynamic Analyzers

A. Understand OWASP and Dynamic Analyzers
B. Implement OWASP Security Coding Practices
C. Understand compliance for code bases

XLVIII. Security Monitoring and Governance

A. Configure Microsoft Defender for Cloud
B. Understand Azure policies
C. Describe initiatives, resource locks and Azure Blueprints
D. Work with Microsoft Defender for Identity

Course Outline (cont'd)

XLIX. Explore package dependencies
- A. Define dependency management strategy
- B. Identify dependencies
- C. Describe elements and componentization of a dependency management
- D. Scan your codebase for dependencies

L. Understand package management
- A. Implement package management
- B. Manage package feed
- C. Consume and create packages
- D. Publish packages

LI. Migrate consolidating and secure artifacts
- A. Identify artifact repositories
- B. Migrate and integrate artifact repositories
- C. Secure package feeds
- D. Understand roles, permissions and authentication

LII. Implement a versioning strategy
- A. Implement a versioning strategy
- B. Promote packages
- C. Push packages from pipeline
- D. Describe semantic and explore best practices for versioning

LIII. Introduction to GitHub Packages
- A. Publish packages
- B. Install packages
- C. Delete and restore packages
- D. Configure access control and visibility

LIV. Implement tools to track usage and flow
- A. Implement tools to track feedback
- B. Plan for continuous monitoring
- C. Implement Application Insights
- D. Use Kusto Query Language (KQL)

LV. Develop monitor and status dashboards
- A. Configure Azure Dashboards
- B. Work with View Designer in Azure Monitor
- C. Create Azure Monitor Workbooks
- D. Monitor with Power BI

LVI. Share knowledge within teams
- A. Share knowledge with development teams
- B. Work with Azure DevOps Wikis
- C. Integrate with Azure Boards

LVII. Design processes to automate application analytics
- A. Automate application analytics
- B. Assist DevOps with rapid responses and augmented search
- C. Integrate telemetry
- D. Implement monitoring tools and technologies

LVIII. Manage alerts, blameless retrospectives and a just culture
- A. Carry out blameless retrospectives and create a just culture
- B. Improve application performance
- C. Explain server response time degradation
- D. Reduce meaningless and non-actionable alerts