AZ-400T00: Designing and Implementing Microsoft DevOps solutions

Elements of this syllabus are subject to change.

About this course

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.

Audience Profile

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.

Introduction to DevOps

This module explores the key areas that organizations must apply to start their DevOps transformation Journey, change the team's mindset, and define timelines and goals.

Learning objectives

By the end of this module, you'll be able to:

- Understand what DevOps is and the steps to accomplish it
- Identify teams to implement the process
- Plan for the transformation with shared goals and timelines
- Plan and define timelines for goals

Choose the right project

This module helps organizations decide the projects to start applying the DevOps process and tools to minimize initial resistance.

Learning objectives

By the end of this module, you'll be able to:

- Understand different projects and systems to guide the journey
- Select a project to start the DevOps transformation
- Identify groups to minimize initial resistance
- Identify project metrics and Key Performance Indicators (KPI's)

Describe team structures

This module explores agile development practices and helps to define and to configure teams and tools for collaboration.

Learning objectives

By the end of this module, you'll be able to:

- Understand agile practices and principles of agile development
- Create a team and agile organizational structure
- Identify ideal DevOps team members
- Select and configure tools for collaboration

Choose the DevOps tools

This module explores Azure DevOps and GitHub tools and helps organizations define their work management tool and licensing strategy.

Learning objectives

By the end of this module, you'll be able to:

- Design a tool integration strategy
- Design a license management strategy (e.g. Azure DevOps and GitHub users)
- Design a strategy for end-to-end traceability from work items to working software
- Design an authentication and access strategy
- Design a strategy for integrating on-premises and cloud resources

Plan Agile with GitHub Projects and Azure Boards

This module introduces you to GitHub Projects, GitHub Project Boards and Azure Boards. It explores ways to link Azure Boards and GitHub, configure GitHub Projects and Project views, and manage work with GitHub Projects.

Learning objectives

By the end of this module, you'll be able to:

- Describe GitHub Projects and Azure Boards
- Link Azure Boards and GitHub
- Configure and Manage GitHub Projects and boards
- Customize Project views

Introduction to source control

This module introduces you to the basics of source control, exploring benefits and best practices.

Learning objectives

By the end of this module, you'll be able to:

- Understand source control
- Apply best practices for source control
- Describe the benefits of using source control

Describe types of source control systems

This module describes different source control systems such Git and TFVC and helps with the initial steps for Git utilization.

Learning objectives

By the end of this module, you'll be able to:

- Apply source control practices in your development process
- Explain differences between centralized and distributed version control
- Understand Git and TFVC
- Develop using Git

Work with Azure Repos and GitHub

This module introduces you to Azure Repos and GitHub and explores ways to migrate from TFVC to Git, and work with GitHub Codespaces for development.

Learning objectives

By the end of this module, you'll be able to:

- Describe Azure Repos and GitHub
- Migrate from TFVC to Git
- Work with GitHub Codespaces

Structure your Git Repo

This module examines Git repositories structure, explains the differences between mono versus multiple repos, and helps you create a changelog.

Learning objectives

By the end of this module, you'll be able to:

- Understand Git repositories
- Implement mono repo or multiple repos
- Explain how to structure Git Repos
- Implement a change log

Manage Git branches and workflows

This module explores Git branching types, concepts, and models for the continuous delivery process. It helps companies defining their branching strategy and organization.

Learning objectives

By the end of this module, you'll be able to:

- Describe Git branching workflows
- Implement feature branches
- Implement GitHub Flow
- Fork a repo

Collaborate with pull requests in Azure Repos

This module presents pull requests for collaboration and code reviews using Azure DevOps and GitHub mobile for pull request

approvals. It helps understanding how pull requests works and how to configure them.

Learning objectives

By the end of this module, you'll be able to:

- Leverages pull requests for collaboration and code reviews
- Give feedback using pull requests
- Configure branch policies
- Use GitHub mobile for pull requests approvals

Identify technical debt

This module examines technical debt, complexity, quality metrics, and plans for effective code reviews and code quality validation.

Learning objectives

By the end of this module, you'll be able to:

- Identify and manage technical debt
- Integrate code quality tools
- Plan code reviews
- Describe complexity and quality metrics

Explore Git hooks

This module describes Git hooks and their usage during the development process, implementation, and behaviour.

Learning objectives

By the end of this module, you'll be able to:

- Understand Git hooks
- Identify when used Git hooks
- Implement Git hooks for automation
- Explain Git hooks' behaviour

Plan foster inner source

This module explains how to use Git to foster inner sources across the organization, implement Fork and its workflows.

Learning objectives

By the end of this module, you'll be able to:

- Use Git to foster inner source across the organization
- Implement fork workflow
- Choose between branches and forks
- Share code between forks

Manage Git repositories

This module explores how to work with large repositories, purge repository data and manage and automate release notes using GitHub.

Learning objectives

By the end of this module, you'll be able to:

- Understand large Git repositories
- Explain VFS for Git
- Use Git Large File Storage (LFS)
- Purge repository data
- Manage and Automate Release Notes with GitHub

Explore Azure Pipelines

This module introduces Azure Pipelines concepts and explains key terms and components of the tool, helping you decide your pipeline strategy and responsibilities.

Learning objectives

By the end of this module, you'll be able to:

- Describe Azure Pipelines
- Explain the role of Azure Pipelines and its components
- Decide Pipeline automation responsibility
- Understand Azure Pipeline key terms

Manage Azure Pipeline agents and pools

This module explores the differences between Microsoft-hosted and self-hosted agents, details job types, and configures agent pools. Understand typical situations to use agent pools and how to manage their security.

Learning objectives

By the end of this module, you'll be able to:

- Choose between Microsoft-hosted and self-hosted agents
- Install and configure Azure Pipelines Agents
- Configure agent pools
- Make the agents and pools secure
- Explore communication to deploy using Azure Pipelines

Describe pipelines and concurrency

This module describes parallel jobs and how to estimate their usage. Also, it presents Azure Pipelines for open-source projects, explores Visual Designer and YAML pipelines.

Learning objectives

By the end of this module, you'll be able to:

- Use and estimate parallel jobs
- Use Azure Pipelines for open-source or private projects
- Use Visual Designer
- Work with Azure Pipelines and YAML

Explore continuous integration

This module details continuous integration practice and the pillars for implementing it in the development lifecycle, its benefits, and properties.

Learning objectives

By the end of this module, you'll be able to:

- Explain why continuous integration matters
- Implement continuous integration using Azure Pipelines
- Explain benefits of continuous integration
- Describe build properties

Implement a pipeline strategy

This module describes pipeline strategies, configuring them, implementing multi-agent builds, and what source controls Azure Pipelines supports.

Learning objectives

By the end of this module, you'll be able to:

- Define a build strategy
- Explain and configure demands
- Implement multi-agent builds
- Use different source control types available in Azure Pipelines

Integrate with Azure Pipelines

This module details Azure Pipelines anatomy and structure, templates, YAML resources, and how to use multiple repositories in your pipeline.

Learning objectives

By the end of this module, you'll be able to:

- Describe advanced Azure Pipelines anatomy and structure
- Detail templates and YAML resources
- Implement and use multiple repositories

Introduction to GitHub Actions

In this module, you will learn what GitHub Actions, action flow, and its elements are. Understand what events are, explore jobs and runners, and how to read console output from actions.

Learning objectives

By the end of this module, you'll be able to:

- Explain GitHub Actions and workflows
- Create and work with GitHub Actions and Workflows
- Describe Events, Jobs and Runners
- Examine output and release management for actions

Learn continuous integration with GitHub Actions

This module details continuous integration using GitHub Actions and describes environment variables, artifacts, best practices, and how to secure your pipeline using encrypted variables and secrets.

Learning objectives

By the end of this module, you'll be able to:

- Implement Continuous Integration with GitHub Actions
- Use environment variables
- Share artifacts between jobs and use Git tags
- Create and manage secrets

Design a container build strategy

This module helps you plan a container build strategy, explains containers and their structure, introduces Docker, microservices, Azure Container Registry, and related services.

Learning objectives

By the end of this module, you'll be able to:

- Design a container strategy
- Work with Docker Containers
- Create an Azure Container Registry
- Explain Docker microservices and containers

Introduction to continuous delivery

This module introduces continuous delivery concepts and their implementation in a traditional IT development cycle.

Learning objectives

By the end of this module, you'll be able to:

- Explain continuous delivery (CD)
- Implement continuous delivery in your development cycle
- Understand releases and deployment
- Identify project opportunities to apply CD

Create a release pipeline

This module describes Azure Pipelines capabilities, build and release tasks.

Learning objectives

By the end of this module, you'll be able to:

- 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
- Implement release jobs

Explore release recommendations

This module explores the critical release strategy recommendations that organizations must consider when designing automated deployments and explains how to define components of a release pipeline and artifact sources, create approves, and configure release gates.

Learning objectives

By the end of this module, you'll be able to:

- Explain things to consider when designing your release strategy
- Define the components of a release pipeline and use artifact sources
- Create a release approval plan
- Implement release gates

Provision and test environments

This module details target environment provisioning, service connections creation process, and test infrastructure setup. Learn how to configure functional test automation and run availability tests.

Learning objectives

By the end of this module, you'll be able to:

- Provision and configure target environment
- Deploy to an environment securely using a service connection
- Configure functional test automation and run availability tests
- Setup test infrastructure

Manage and modularize tasks and templates

This module describes the creation of task and variable groups and using release variables and stage variables in your pipeline.

Learning objectives

By the end of this module, you'll be able to:

- Use and manage task and variable groups
- Use release variables and stage variables in your release pipeline
- Use variables in release pipelines

Automate inspection of health

This module describes how to automate the inspection of health events, configure notifications in Azure DevOps and GitHub, set up service hooks to monitor pipelines, measure the quality of your release process, and detail release gates for quality purposes. You'll examine release management tools and details about them.

Learning objectives

By the end of this module, you'll be able to:

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

Introduction to deployment patterns

This module introduces deployment patterns and explains microservices architecture to help improve the deployment cycle and examine classical and modern deployment patterns.

Learning objectives

- Describe deployment patterns
- Explain microservices architecture
- Understand classical and modern deployment patterns
- Plan and design your architecture

Implement blue-green deployment and feature toggles

This module describes the blue-green deployment process and introduces feature toggle techniques to implement in the development process.

Learning objectives

By the end of this module, you'll be able to:

- Explain deployment strategies
- Implement blue green deployment
- Understand deployment slots
- Implement and manage feature toggles

Implement canary releases and dark launching

This module describes deployment strategies around canary releases and dark launching and examines traffic managers.

Learning objectives

By the end of this module, you'll be able to:

- Describe deployment strategies
- Implement canary release
- Explain traffic manager
- Understand dark launching

Implement A/B testing and progressive exposure deployment

This module introduces A/B test and progressive exposure deployment concepts and explores CI/CD with deployment rings -- ring-based deployment.

Learning objectives

By the end of this module, you'll be able to:

- Implement progressive exposure deployment
- Implement A/B testing
- Implement CI/CD with deployment rings
- Identify the best deployment strategy

Integrate with identity management systems

This module describes the integration with GitHub and single sign-on (SSO) for authentication, service principals, and managed service identities.

Learning objectives

By the end of this module, you'll be able to:

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

Manage application configuration data

This module explores ways to rethink application configuration data and the separation of concerns method. Explore Azure App Configuration, details Key-value pairs, App Configuration feature management, and integrate Azure Key Vault with Azure Pipelines.

Learning objectives

By the end of this module, you'll be able to:

- Rethink application configuration data
- Understand separation of concerns
- Integrate Azure Key Vault with Azure Pipelines
- Manage secrets, tokens and certificates
- Describe Azure App Configuration
- Understand Key-value pairs
- Understand app configuration feature management
- Implement application configuration

Explore infrastructure as code and configuration management

This module describes key concepts of infrastructure as code and environment deployment creation and configuration. Also, understand the imperative, declarative, and idempotent configuration and how it applies to your company.

Learning objectives

By the end of this module, you'll be able to:

- Understand how to deploy your environment
- Plan your environment configuration

- Choose between imperative versus declarative configuration
- Explain idempotent configuration

Create Azure resources using Azure Resource Manager templates

This module explores Azure Resource Manager templates and their components and details dependencies and modularized templates with secrets.

Learning objectives

By the end of this module, you'll be able to:

- Create Azure resources using Azure Resource Manager templates
- Understand Azure Resource Manager templates and template components
- Manage dependencies and secrets in templates
- Organize and modularize templates

Create Azure resources by using Azure CLI

This module explains Azure CLI to create Azure resources, run templates, and detail Azure CLI commands.

Learning objectives

By the end of this module, you'll be able to:

- Create Azure resources using Azure CLI
- Understand and work with Azure CLI
- Run templates using Azure CLI
- Explains Azure CLI commands

Explore Azure Automation with DevOps

This module describes Azure Automation with Azure DevOps, using runbooks, webhooks, and PowerShell workflows. You'll learn how to create and manage automation for your environment.

Learning objectives

By the end of this module, you'll be able to:

- Implement automation with Azure DevOps
- Create and manage runbooks
- Create webhooks
- Create and run a workflow runbook and PowerShell workflows
- Plan for hybrid management

Implement Desired State Configuration (DSC)

This module describes Desired State Configuration (DSC) and its components for implementation. You can exercise how to import, compile and automate your environment creation, and use DSC for Linux automation on Azure.

Learning objectives

By the end of this module, you'll be able to:

- Implement Desired State Configuration (DSC)
- Describe Azure Automation State Configuration
- Implement DSC and Linux Automation on Azure
- Plan for hybrid management

Implement Bicep

This module explains Bicep and how it integrates with different tools such as Azure CLI and Visual Studio Code for environment deployment configuration.

Learning objectives

By the end of this module, you'll be able to:

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

Introduction to Secure DevOps

This module introduces DevSecOps concepts, SQL injection attacks, threat modeling, and security for continuous integration.

Learning objectives

By the end of this module, you'll be able to:

- Identify SQL injection attack
- Understand DevSecOps
- Implement pipeline security
- Understand threat modelling

Implement open-source software

This module explores open-source software and corporate concerns with software components. Also, it explains common open-source licenses, license implications, and ratings.

Learning objectives

By the end of this module, you'll be able to:

- Implement open-source software
- Explain corporate concerns for open-source components
- Describe open-source licenses
- Understand the license implications and ratings

Software Composition Analysis

This module explains Composition Analysis, how to inspect and validate code bases for compliance, integration with security tools, and integration with Azure Pipelines.

Learning objectives

By the end of this module, you'll be able to:

- Inspect and validate code bases for compliance
- Integrate security tools like WhiteSource with Azure DevOps
- Implement pipeline security validation
- Interpret alerts from scanning tools
- Configure GitHub Dependabot alerts and security

Static analyzers

This module introduces the static analyzers SonarCloud and CodeQL in GitHub.

Learning objectives

By the end of this module, you'll be able to:

- Understand Static Analyzers
- Work with SonarCloud
- Work with CodeQL in GitHub
- Interpret alerts from scanning tools

OWASP and Dynamic Analyzers

This module explores OWASP and Dynamic Analyzers for penetration testing, results, and bugs.

Learning objectives

- By the end of this module, you'll be able to:
- Understand OWASP and Dynamic Analyzers
- Implement OWASP Security Coding Practices
- Understand compliance for code bases

Security Monitoring and Governance

This module describes security monitoring and governance with Microsoft Defender for Cloud and its usage scenarios, Azure Policies, Microsoft Defender for Identity, and security practices related to the tools.

Learning objectives

By the end of this module, you'll be able to:

- Configure Microsoft Defender for Cloud
- Understand Azure policies
- Describe initiatives, resource locks and Azure Blueprints
- Work with Microsoft Defender for Identity

Explore package dependencies

This module explores dependency management concepts and helps to identify project dependencies. You will learn how to decompose your system, identify dependencies, and package componentization.

Learning objectives

By the end of this module, you'll be able to:

- Define dependency management strategy
- Identify dependencies
- Describe elements and componentization of a dependency management
- Scan your codebase for dependencies

Understand package management

This module describes package feeds, common public package sources, and how to create and publish packages.

Learning objectives

By the end of this module, you'll be able to:

- Implement package management
- Manage package feed
- Consume and create packages
- Publish packages

Migrate consolidating and secure artifacts

This module details package migration, consolidation, and configuration to secure access to package feeds and artifact repositories.

Learning objectives

By the end of this module, you'll be able to:

- Identify artifact repositories
- Migrate and integrate artifact repositories
- Secure package feeds
- Understand roles, permissions and authentication

Implement a versioning strategy

This module explains versioning strategies for packaging, best practices for versioning, and package promotion.

Learning objectives
By the end of this module, you'll be able to:

- Implement a versioning strategy
- Promote packages
- Push packages from pipeline
- Describe semantic and explore best practices for versioning

Introduction to GitHub Packages

This module introduces you to GitHub Packages. It explores ways to control permissions and visibility, publish, install, delete and restore packages using GitHub.

Learning objectives
By the end of this module, you'll be able to:

- Publish packages
- Install packages
- Delete and restore packages
- Configure access control and visibility

Implement tools to track usage and flow

This module introduces you to continuous feedback practices and tools to track usage and flow, such as Azure Logs Analytics, Kusto Query Language (KQL), and Application Insights.

Learning objectives
By the end of this module, you'll be able to:

- Implement tools to track feedback
- Plan for continuous monitoring
- Implement Application Insights
- Use Kusto Query Language (KQL)

Develop monitor and status dashboards

This module explains steps to develop monitoring with Azure Dashboards, work with View Designer and Azure Monitor, and create Azure Monitor Workbooks. Also, explore tools to supports monitoring with Power BI.

Learning objectives

By the end of this module, you'll be able to:

- Configure Azure Dashboards
- Work with View Designer in Azure Monitor
- Create Azure Monitor Workbooks
- Monitor with Power BI

Share knowledge within teams

This module describes how to share knowledge within teams, Azure DevOps Wikis, and integration with Azure Boards.

Learning objectives

By the end of this module, you'll be able to:

- Share knowledge with development teams
- Work with Azure DevOps Wikis
- Integrate with Azure Boards

Design processes to automate application analytics

This module helps designing process to Application Insights, explores telemetry and monitoring tools and technologies.

Learning objectives

By the end of this module, you'll be able to:

- Automate application analytics
- Assist DevOps with rapid responses and augmented search
- Integrate telemetry
- Implement monitoring tools and technologies

Manage alerts, blameless retrospectives and a just culture

This module examines alerts, blameless retrospectives and creates a just culture. It helps improving application performance, reducing meaningless and non-actionable alerts, and explains server response-time degradation.

Learning objectives

By the end of this module, you'll be able to:

- Carry out blameless retrospectives and create a just culture
- Improve application performance
- Explain server response time degradation
- Reduce meaningless and non-actionable alerts