

# MS-AZ400T00: DESIGNING AND IMPLEMENTING MICROSOFT DEVOPS SOLUTIONS



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
4 Days	Advanced	Azure	Instructor-led	NA

## INTRODUCTION

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 implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam.

## PREREQUISITES

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

- Experience with Azure: You should have a solid understanding of Azure administration and development.
- Knowledge of DevOps practices: Familiarity with DevOps processes and tools is essential.
- Prior certification: It's recommended to have an Azure Administrator Associate or Azure Developer Associate certification before attempting the AZ-400 exam.

## COURSE OBJECTIVES

After completion of this course, you will be able to:

- Develop an instrumentation strategy
- Develop a Site Reliability Engineering (SRE) strategy
- Develop a security and compliance plan
- Manage source control
- Facilitate communication and collaboration
- Define and implement continuous integration

## COURSE CONTENT

### Module 1: 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

- 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.

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.

### Module 2: 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

- Introduction.
- Introduction to GitHub Projects and Project boards.
- Introduction to Azure Boards.
- Link GitHub to Azure Boards.
- Configure GitHub Projects.
- Manage work with GitHub Project boards.
- Customize Project views.
- Collaborate using team discussions.
- Agile Plan and Portfolio Management with Azure Boards.
- Knowledge check.



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.

### Module 3: 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

- Introduction
- Explore branch workflow types.
- Explore feature branch workflow.
- Explore Git branch model for continuous delivery.
- Explore GitHub flow.
- Explore fork workflow.
- Version Control with Git in Azure Repos.
- Knowledge check.

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

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

### Module 4: 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

- Introduction
- Collaborate with pull requests
- Exercise - Azure Repos collaborating with pull requests
- Examine GitHub mobile for pull request approvals
- Knowledge check

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.

### Module 5: Explore Git hooks

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

Learning objectives

- Introduction
- Introduction to Git hooks
- Implement Git hooks
- Knowledge check

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.

### Module 6: Plan foster inner source

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

Learning objectives

Introduction

- Explore foster inner source.
- Implement the fork workflow.
- Describe inner source with forks.
- Knowledge check.

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.

### Module 7: Identify technical debt

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

Learning objectives

- Introduction.
- Examine code quality.
- Examine 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.
- Knowledge check.

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.

### Module 8: 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

- Introduction.
- Explore the concept of pipelines in DevOps.
- Describe Azure Pipelines.
- Understand Azure Pipelines key terms.
- Knowledge check.

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.

### Module 9: 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

- Introduction.
- Choose between Microsoft-hosted versus self-hosted agents.
- Explore job types.
- Introduction to agent pools.
- Explore predefined agent pool.
- Understand typical situations for agent pools.
- Communicate with Azure Pipelines.
- Communicate to deploy to target servers.
- Examine other considerations.
- Describe security of agent pools.
- Configure agent pools and understand pipeline styles.
- Knowledge check.

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.

### Module 10: Describe pipelines and concurrency

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

- Introduction.
- Understand parallel jobs.
- Estimate parallel jobs.
- Describe Azure Pipelines and open-source projects.
- Explore Azure Pipelines and Visual Designer.
- Describe Azure Pipelines and YAML.
- Knowledge check.

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.

## Module 11: Implement a pipeline strategy

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

Learning objectives

- Introduction.
- Configure agent demands.
- Implement multi-agent builds.
- Explore source control types supported by Azure Pipelines.
- Knowledge check.

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.

## Module 12: 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

- Introduction.
- Describe the anatomy of a pipeline.
- Understand the pipeline structure.
- Detail templates.
- Explore YAML resources.
- Use multiple repositories in your pipeline.
- Knowledge check.

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.

## Module 13: Introduction to GitHub Actions

In this module, you 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

- Introduction.
- What are actions?
- Explore Actions flow.
- Understand Workflows.
- Describe standard workflow syntax elements.
- Explore Events.
- Explore Jobs.
- Explore Runners.
- Examine release and test an action.
- Knowledge check.

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.

## Module 14: 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

- Introduction.
- Describe continuous integration with actions.
- Examine environment variables.
- Share artifacts between jobs.
- Examine Workflow badges.
- Describe best practices for creating actions.
- Mark releases with Git tags.
- Create encrypted secrets.
- Use secrets in a workflow.
- Implement GitHub Actions for CI/CD.
- Knowledge check.

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.

## Module 15: 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

- Introduction.
- Examine structure of containers.
- Work with Docker containers.
- Understand Dockerfile core concepts.
- Examine multi-stage dockerfiles.
- Examine considerations for multiple stage builds.
- Explore Azure container-related services.
- Deploy Docker containers to Azure App Service web apps.
- Knowledge check.

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.

## Module 16: Create a release pipeline

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

- Introduction
- Describe Azure DevOps release pipeline capabilities.
- Explore release pipelines.
- Explore artifact sources.
- Choose the appropriate artifact source.
- Exercise - select an artifact source.
- Examine considerations for deployment to stages.
- Exercise - set up stages.
- Explore build and release tasks.
- Explore custom build and release tasks.
- Explore release jobs.
- Configure Pipelines as Code with YAML.
- Knowledge check.

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.

## Module 17: 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 approvals, and configure release gates.

Learning objectives

- Introduction.
- Understand the delivery cadence and three types of triggers.
- Exercise - select your delivery and deployment cadence.
- Explore release approvals.
- Exercise - set up manual approvals.
- Explore release gates.
- Use release gates to protect quality.
- Control Deployments using Release Gates.
- Knowledge check.

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.

## Module 18: 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 approvals, and configure release gates.

Learning objectives

- Introduction.
- Understand the delivery cadence and three types of triggers.
- Exercise - select your delivery and deployment cadence.
- Explore release approvals.
- Exercise - set up manual approvals.
- Explore release gates.
- Use release gates to protect quality.
- Control Deployments using Release Gates.
- Knowledge check.

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.

## Module 19: 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

- Introduction.
- Provision and configure target environments.
- Exercise - set up service connections.
- Configure automated integration and functional test automation.
- Understand Shift-left.
- Set up and run availability tests.
- Explore Azure Load Testing.
- Set up and run functional tests.
- Knowledge check.

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.

## Module 20: 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

- Introduction.
- Examine task groups.
- Exercise - create and manage task groups.
- Explore variables in release pipelines.
- Understand variable groups.
- Exercise - create and manage variable groups.
- Knowledge check.

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.

## Module 21: 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

- Introduction.
- Automate inspection of health.
- Explore events and notifications.
- Explore service hooks.
- Exercise - set up service hooks to monitor the pipeline.
- Configure Azure DevOps notifications.
- Configure GitHub notifications.
- Explore how to measure quality of your release process.
- Examine release notes and documentation.
- Examine considerations for choosing release management tools.
- Explore common release management tools.
- Knowledge check.

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.

## Module 22: 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

- Introduction.
- Explore microservices architecture.
- Examine classical deployment patterns.
- Understand modern deployment patterns.

Knowledge check.

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

- Explain microservices architecture.
- Understand classical and modern deployment patterns.



- Plan and design your architecture.

## Module 23: 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

- Introduction.
- What is blue-green deployment?
- Explore deployment slots.
- Exercise - set up a blue-green deployment.
- Introduction to feature toggles.
- Describe feature toggle maintenance.
- Knowledge check.

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.

## Module 24: Implement canary releases and dark launching

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

Learning objectives

- Introduction.
- Explore canary releases.
- Examine traffic manager.
- Understand dark launching.
- Knowledge check.

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

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

## Module 25: 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

- Introduction.
- What is A/B testing?
- Explore CI-CD with deployment rings.
- Exercise - Ring-based deployment.
- Knowledge check.

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.

## Module 26: 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

- Introduction.
- Integrate GitHub with single sign-on. (SSO)
- Explore service principals.
- Explore Managed Identity.
- Knowledge check.

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.

## Module 27: 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

- Introduction.
- Rethink application configuration data.
- Explore separation of concerns
- Understand external configuration store patterns.
- Introduction to Azure App Configuration.
- Examine Key-value pairs.
- Examine App configuration feature management.
- Integrate Azure Key Vault with Azure Pipelines.
- Manage secrets, tokens and certificates.
- Examine DevOps inner and outer loop.
- Integrate Azure Key Vault with Azure DevOps.
- Enable Dynamic Configuration and Feature Flags.
- Knowledge check.

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.

## Module 28: 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

- Introduction.
- Explore environment deployment.
- Examine environment configuration.
- Understand imperative versus declarative configuration.
- Understand idempotent configuration.
- Knowledge check.

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.

## Module 29: 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

- Introduction.
- Why use Azure Resource Manager templates?
- Explore template components.
- Manage dependencies.
- Modularize templates.
- Manage secrets in templates.
- Deployments using Azure Bicep templates.
- Knowledge check.

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.

## Module 30: 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

- Introduction.
- What is Bicep?
- Install Bicep.
- Exercise - Create Bicep templates.
- Understand Bicep file structure and syntax.
- Exercise - Deploy a Bicep file from Azure Pipelines.
- Exercise - Deploy a Bicep file from GitHub workflows.
- Knowledge check.

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.

## Module 31: 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

- Introduction.
- What is Azure CLI?
- Work with Azure CLI.
- Exercise - Run templates using Azure CLI.
- Knowledge check.

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.

## Module 32: 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

## Module 33: 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

## Module 34: 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

## Module 35: 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

## Module 36: Manage application configuration data

This module explores ways to rethink application configuration data and the separation of

concerns method. It helps you understand configuration patterns and how to integrate Azure Key Vault with Azure Pipelines. Also, describes Azure App Configuration and details Key-value pairs and App Configuration feature management.

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

## Module 37: 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

## Module 38: 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

## Module 39: 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

## Module 40: 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

## Module 41: 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

## Module 42: 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

## Module 43: 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

## Module 44: 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

## Module 45: 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

## Module 46: 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

## Module 47: 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

## Module 48: 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

## Module 49: 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

## Module 50: 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

## Module 51: 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

## Module 52: 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

## Module 53: 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

## Module 54: 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)

## Module 55: 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

## Module 56: 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

## Module 57: 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

## Module 58: 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

## ASSOCIATED CERTIFICATIONS & EXAM

This course will prepare delegates to write the Microsoft AZ-400: Designing and Implementing Microsoft DevOps Solutions exam.