

COURSE OUTLINE



Course Name: Designing And Implementing Microsoft DevOps Solutions

Course Code: MS-AZ400T00

DURATION	SKILL LEVEL	DELIVERY METHOD	TRAINING CREDITS	TECHNOLOGY
4 Day	Intermediate	VIL/TILT	N/A	DevOps

Course Overview

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

Target Audience

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

Pre-requisites

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:

- MS-AZ900T00: Azure Fundamentals
- MS-AZ104T00: Microsoft Azure Administrator
- MS-AZ204: Developing Solutions for Microsoft Azure

Course Objective

After completing this course, students will be able to:

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and Key Performance Indicators (KPI's)
- Create a team and agile organizational structure
- 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
- Describe the benefits of using Source Control
- Describe Azure Repos and GitHub
- Migrate from TFVC to Git
- Manage code quality including technical debt SonarCloud, and other tooling solutions
- Build organizational knowledge on code quality
- Explain how to structure Git repos
- Describe Git branching workflows
- Leverage pull requests for collaboration and code reviews
- Leverage Git hooks for automation
- Use Git to foster inner source across the organization
- Explain the role of Azure Pipelines and its components
- Configure Agents for use in Azure Pipelines
- Explain why continuous integration matters
- Implement continuous integration using Azure Pipelines
- Define Site Reliability Engineering
- Design processes to measure end-user satisfaction and analyze user feedback
- Design processes to automate application analytics
- Manage alerts and reduce meaningless and non-actionable alerts
- Carry out blameless retrospectives and create a just culture
- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure
- Describe the potential challenges with integrating open-source software
- Inspect open-source software packages for security and license compliance
- Manage organizational security and compliance policies
- Integrate license and vulnerability scans into build and deployment pipelines
- Configure build pipelines to access package security and license ratings

Course Content

Module 1: Get started on a DevOps transformation journey

Lesson

- Introduction to DevOps
- Choose the right project
- Describe team structures
- Migrate to DevOps
- Introduction to source control
- Describe types of source control systems
- Work with Azure Repos and GitHub

Lab : Agile planning and portfolio management with Azure Boards

Lab : Version controlling with Git in Azure Repos

Module 2: Work with Git for enterprise DevOps

Lesson

- Structure your Git Repo
- Manage Git branches and workflows
- Collaborate with pull requests in Azure Repos
- Explore Git hooks
- Plan fostering inner source
- Manage Git repositories

Lab : Version controlling with Git in Azure Repos

Module 3: Implement CI with Azure Pipelines and GitHub Actions

Lesson

- 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
-

Lab : Configuring agent pools and understanding pipeline styles

Module 4: Design and implement a release strategy

Lesson

- Introduction to continuous delivery
- Explore release strategy recommendations
- Build a high-quality release pipeline
- 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

Lab : Controlling deployments using Release Gates

Lab : Creating a release dashboard

Lab : Feature flag management with LaunchDarkly and Azure DevOps

Module 5: Implement a secure continuous deployment using Azure Pipelines

Lesson

- Create a release pipeline
- Configure and provision environments
- Manage and modularize tasks and templates
- Automate inspection of health
- Introduction to security development process
- Manage application configuration data
- Integrate with identity management systems
- Implement application configuration

Lab : Integrating Azure Key Vault with Azure DevOps

Lab : Setting up and running functional tests

Lab : Configuring pipelines as code with YAML

Module 6: Manage infrastructure as code using Azure, DSC, and third-party tools

Lesson

- 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)
- Introduction to Chef and Puppet
- Implement Ansible
- Implement Terraform

Lab : Deployments using Azure Resource Manager templates

Lab : Ansible with Azure

Lab : Automating infrastructure deployments in the cloud with Terraform and Azure Pipelines

Module 7: Design and implement a dependency management strategy

Lesson

- Explore package dependencies
- Understand package management
- Migrate, consolidating and secure artifacts
- Implement a versioning strategy

Lab : Package management with Azure Artifacts

Module 8: Create and manage containers using Docker and Kubernetes

Lesson

- Design a container build strategy
 - Implement Docker multi-stage builds
 - Implement Azure Kubernetes Service (AKS)
 - Explore Kubernetes tooling
 - Integrate AKS with Pipelines
-

Module 9: Implement continuous feedback

Lesson

- Implement tools to track usage and flow
- Implement route for mobile application crash report data
- Develop monitor and status dashboards
- § Share knowledge within teams
- Explore SRE and design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback
- Design processes to automate application analytics
- Manage alerts, Blameless retrospectives and a just culture

Lab : Integration between Azure DevOps and Microsoft Teams

Lab : Monitoring application performance with Application Insights

Module 10: Implement security and validate code bases for compliance

Lesson

- Understand security in the Pipeline
- Introduction to Azure Security Center
- Implement open-source software
- Manage anti-malware and anti-spam policies
- Integrate license and vulnerability scans
- Identify technical debt

Lab : Implement security and compliance in Azure DevOps Pipelines

Lab : Managing technical debt with SonarQube and Azure DevOps

Follow on Course

[Schedules | Netcampus Group](#)
