



Information Technology

AZ-400T00-Designing and Implementing Microsoft DevOps Solutions

Course Introduction

This training course is designed to provide participants the essential concepts and skills needed for them to design and implement DevOps processes and practices. This course will enable them 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

- DevOps Engineers
- Software Developers
- System Administrators
- IT Professionals
- Project Managers
- Cloud Architects
- Development Teams
- Quality Assurance Engineers
- Technical Leads and Architects

Learning Objectives

- Plan for the transformation with shared goals and timelines
- Choose a project and identify project metrics and Key Performance Indicators (KPI's)
- Create a team and agile organizational structure
- Design a tool integration strategy, license management strategy (e.g., Azure DevOps and GitHub users), strategy for end-to-end traceability from work items to working

software, authentication and access strategy, and strategy for integrating on-premises and cloud resources

- Recognize 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 Outline

- Day 01

- Module 1: Planning for DevOps**

- Transformation Planning

- Project Selection
- Team Structures
- Migrating to Azure DevOps
- Lab: Agile Planning and Portfolio Management with Azure Boards

Module 2: Getting Started with Source Control

- What is Source Control
- Benefits of Source Control
- Types of Source Control Systems
- Introduction to Azure Repos
- Introduction to GitHub
- Migrating from Team Foundation Version Control (TFVC) to Git in Azure Repos
- Lab: Version Controlling with Git in Azure Repos

Module 3: Managing Technical Debt

- Identifying Technical Debt
- Knowledge Sharing within Teams
- Modernizing Development Environments with Codespaces
- Lab: Sharing Team Knowledge using Azure Project Wikis

Module 4: Working with Git for Enterprise DevOps

- How to Structure Your Git Repo
- Git Branching Workflows
- Collaborating with Pull Requests in Azure Repos
- Why Care About Git Hooks
- Fostering Inner Source
- Managing Git Repositories
- Lab: Version Controlling with Git in Azure Repos

Module 5: Configuring Azure Pipelines

- The Concept of Pipelines in DevOps
- Azure Pipelines
- Evaluate use of Hosted versus Self-Hosted Agents
- Agent Pools
- Pipelines and Concurrency
- Azure DevOps and Open-Source Projects (Public Projects)
- Azure Pipelines YAML versus Visual Designer

Lab: Configuring Agent Pools and Understanding Pipeline Styles

- **Day 02**

Module 6: Implementing Continuous Integration using Azure Pipelines

- Continuous Integration Overview
- Implementing a Build Strategy
- Integration with Azure Pipelines
- Integrating External Source Control with Azure Pipelines
- Set Up Self-Hosted Agents
- Lab: Enabling Continuous Integration with Azure Pipelines
- Lab: Integrating External Source Control with Azure Pipelines

Module 7: Managing Application Configuration and Secrets

- Introduction to Security
- Implement a Secure Development Process
- Rethinking Application Configuration Data
- Manage Secrets, Tokens, and Certificates
- Integrating with Identity Management Systems
- Implementing Application Configuration
- Lab: Integrating Azure Key Vault with Azure DevOps

Module 8: Implementing Continuous Integration with GitHub Actions

- GitHub Actions
- Continuous Integration with GitHub Actions
- Securing Secrets for GitHub Actions
- Lab: GitHub Actions Continuous Integration

Module 9: Designing and Implementing a Dependency Management Strategy

- Packaging Dependencies
- Package Management
- Migrating and Consolidating Artifacts
- Package Security

- Implementing a Versioning Strategy
- Lab: Package Management with Azure Artifacts

Module 10: Designing a Release Strategy

- Introduction to Continuous Delivery
- Release Strategy Recommendations
- Building a High-Quality Release pipeline
- Choosing the Right Release Management Tool
- Lab: Controlling Deployments using Release Gates
- Lab: Creating a Release Dashboard

• Day 03

Module 11: Implementing Continuous Deployment using Azure Pipelines

- Create a Release Pipeline
- Provision and Configure Environments
- Manage and Modularize Tasks and Templates
- Configure Automated Integration and Functional Test Automation
- Automate Inspection of Health
- Lab: Configuring Pipelines as Code with YAML
- Lab: Setting up and Running Functional Tests

Module 12: Implementing an Appropriate Deployment Pattern

- Introduction to Deployment Patterns
- Implement Blue Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching
- AB Testing
- Progressive Exposure Deployment
- Lab: Feature Flag Management with LaunchDarkly and Azure DevOps

Module 13: Managing Infrastructure and Configuration using Azure Tools

- Infrastructure as Code and Configuration Management
- Create Azure Resources using ARM Templates
- Create Azure Resources using Azure CLI
- Azure Automation with DevOps
- Desired State Configuration (DSC)
- Lab: Azure Deployments using Resource Manager Templates

Module 14: Third Party Infrastructure as Code Tools Available with Azure

- Chef
- Puppet
- Ansible
- Terraform
- Lab: Automating Infrastructure Deployments in the Cloud with Terraform and Azure Pipelines
- Lab: Deploying Apps with Chef on Azure
- Lab: Deploy App with Puppet on Azure
- Lab: Ansible with Azure

Module 15: Managing Containers using Docker

- Implementing a Container Build Strategy
- Implementing Docker Multi-Stage Builds
- Lab: Modernizing Existing ASP.NET Apps with Azure

• Day 04

Module 16: Creating and Managing Kubernetes Service Infrastructure

- Azure Kubernetes Service
- Kubernetes Tooling
- Integrating AKS with Pipelines
- Lab: Deploying a Multi-Container Application to Azure Kubernetes Service

- Implement Tools to Track System Usage, Feature Usage, and Flow
- Implement Routing for Mobile Application Crash Report Data
- Develop Monitoring and Status Dashboards
- Integrate and Configure Ticketing Systems
- Lab: Monitoring Application Performance with Application Insights

• **Day 05**

Module 18: Implementing System Feedback Mechanisms

- Site Reliability Engineering
- Design Practices to Measure End-User Satisfaction
- Design Processes to Capture and Analyze User Feedback
- Design Processes to Automate Application Analytics
- Managing Alerts
- Blameless Retrospectives and a Just Culture
- Lab: Integration between Azure DevOps and Teams

Module 19: Implementing Security in DevOps Projects

- Security in the Pipeline
- Azure Security Center
- Lab: Implement Security and Compliance in an Azure DevOps Pipeline

Module 20: Validating Code Bases for Compliance

- Open-Source Software
- Managing Security and Compliance Policies
- Integrating License and Vulnerability Scans
- Lab: Managing Technical Debt with SonarQube and Azure DevOps

Confirmed Sessions

FROM	TO	DURATION	FEES	LOCATION
June 23, 2025	June 27, 2025	5 days	4250.00 \$	UAE - Dubai
Sept. 22, 2025	Sept. 26, 2025	5 days	4950.00 \$	England - London
Dec. 15, 2025	Dec. 19, 2025	5 days	4250.00 \$	UAE - Abu Dhabi