

AZ-400T00-A: Designing and Implementing  
Microsoft DevOps solutions  
Course certification training

**Course Agenda**

## **Index**

- Module 1: Planning for DevOps
- Module 2: Getting Started with Source Control
- Module 3: Managing Technical Debt
- Module 4: Working with Git for Enterprise DevOps
- Module 5: Configuring Azure Pipelines
- Module 6: Implementing Continuous Integration using Azure Pipelines
- Module 7: Managing Application Configuration and Secrets
- Module 8: Implementing Continuous Integration with GitHub Actions
- Module 9: Designing and Implementing a Dependency Management Strategy
- Module 10: Designing a Release Strategy
- Module 11: Implementing Continuous Deployment using Azure Pipelines
- Module 12: Implementing an Appropriate Deployment Pattern
- Module 13: Managing Infrastructure and Configuration using Azure Tools
- Module 14: Third-Party Infrastructure as Code Tools Available with Azure
- Module 15: Managing Containers using Docker
- Module 16: Creating and Managing Kubernetes Service Infrastructure
- Module 17: Implementing Feedback for Development Teams
- Module 18: Implementing System Feedback Mechanisms
- Module 19: Implementing Security in DevOps Projects
- Module 20: Validating Code Bases for Compliance

## Course Curriculum

### **Module 1: Planning for DevOps**

Transformation Planning  
Project Selection  
Team Structures  
Migrating to Azure DevOps

### **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

### **Module 3: Managing Technical Debt**

Identifying Technical Debt  
Knowledge Sharing within Teams  
Modernizing Development Environments with Codespaces

### **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

### **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

## **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

## **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

## **Module 8: Implementing Continuous Integration with GitHub Actions**

GitHub Actions  
Continuous Integration with GitHub Actions  
Securing Secrets for GitHub Actions

## **Module 9: Designing and Implementing a Dependency Management Strategy**

Packaging Dependencies  
Package Management  
Migrating and Consolidating Artifacts  
Package Security  
Implementing a Versioning Strategy

## **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

## **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

## **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

## **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)

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

Chef  
Puppet

Ansible  
Terraform

### **Module 15: Managing Containers using Docker**

Implementing a Container Build Strategy  
Implementing Docker Multi-Stage Builds

### **Module 16: Creating and Managing Kubernetes Service Infrastructure**

Azure Kubernetes Service  
Kubernetes Tooling  
Integrating AKS with Pipelines

### **Module 17: Implementing Feedback for Development Teams**

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

### **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

### **Module 19: Implementing Security in DevOps Projects**

Security in the Pipeline  
Azure Security Center

### **Module 20: Validating Code Bases for Compliance**

Open-Source Software  
Managing Security and Compliance Policies  
Integrating License and Vulnerability Scans

