AZ-400T00: Designing and Implementing Microsoft® DevOps Solutions

Duration: 5 Days

Method: Instructor-Led Training (ILT) | Live Online Training

Certification: *Microsoft Certified: DevOps Engineer Expert —* ***Exam:*** *AZ-400: Designing and Implementing Microsoft DevOps Solutions*

Course Description

This course provides the knowledge and skills to design and implement DevOps processes and practices. Participants will learn how to:

- Plan for DevOps,
- Use source control,
- Scale Git for an enterprise,
- Consolidate artefacts,
- 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

This course is intended for persons who are interested in:

- Designing and implementing DevOps processes
- Passing the certification exam.

Prerequisites

To attend this course, candidates must have **either**:

- 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.

  NOTE: *It would be helpful to have experience in an organization that delivers software.*

Prerequisites *Continued*

- Completed the following courses to obtain the required knowledge and understanding of:

Delivery Method	Azure and Cloud Computing:	Azure Administration:	Azure Developer:
Instructor-Led	• *AZ-900T00/T001: Azure Fundamentals*	• *AZ-104T00: Microsoft Azure Administrator* ***AND*** *AZ-010T00: Azure Administration for AWS SysOps*	• *AZ-204 T00: Developing Solutions for Microsoft Azure* ***AND*** *AZ-020 T00: Microsoft Azure Solutions for AWS Developers*
Self-Study	• *Azure Fundamentals*	• *Prerequisites for Azure Administrators*	• *Create Serverless Applications*

- Obtained either the *Microsoft Certified: Azure Administrator Associate* **OR** *Microsoft Certified: Azure Developer Associate* certifications.

Course Objectives

Upon successful completion of this course, attendees will be able to:

- Plan for the transformation with shared goals and timelines.
- Select a project and identify project metrics and Key Performance Indicators (KPIs).
- 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.

Course Objectives *Continued*

- Implement continuous integration using Azure Pipelines.
- Define Site Reliability Engineering.
- Design processes to measure end-user satisfaction and analyse 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 Topics

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 the 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

Course Topics *Continued*

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)

Course Topics *Continued*

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 Analyse 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 Centre

Module 20: Validating Code Bases for Compliance

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

LABS INCLUDED