AZ-400T01-A: Implementing DevOps Development Processes

Module 1: Getting started with Source Control
Lessons
- What is Source Control?
- Benefits of Source Control
- Types of source control systems
- Introduction to Azure Repos
- Migrating from TFVC to Git
- Authenticating to your Git Repos

After completing this module, students will be able to:
- Describe the benefits of using source control
- Migrate from TFVC to Git

Module 2: Scaling git for enterprise DevOps
Lessons
- How to structure your git repo
- Git Branching workflows
- Collaborating with Pull Requests
- Why care about GitHooks?
- Fostering Internal Open Source
- Git Version
- Public projects
- Files in Git

After completing this module, students will be able to:
- Scale Git for Enterprise DevOps

Module 3: Implement & Manage Build Infrastructure
Lessons
- The concept of pipelines in DevOps
- Azure Pipelines
- Evaluate use of Hosted vs Private Agents
- Agent pools
- Pipelines & Concurrency
- Azure DevOps and Open Source projects
- Azure Pipelines YAML vs Visual Designer
- Setup private agents
- Integrate Jenkins with Azure Pipelines
- Integration external source control with Azure Pipelines
- Analyze & Integrate Docker multi-stage builds

After completing this module, students will be able to:
- Implement and manage build infrastructure
Module 4: Managing application config & secrets
Lessons
- Introduction to Security
- Implement secure & compliant development process
- Rethinking application config data
- Manage secrets, tokens & certificates
- Implement tools for managing security and compliance in a pipeline

After completing this module, students will be able to:
- Manage application config & secrets

Module 5: Implement a mobile DevOps strategy
Lessons
- Introduction to Mobile DevOps
- Introduction to Visual Studio App Center
- Manage mobile target device sets and distribution groups
- Manage target UI test device sets
- Provision tester devices for deployment
- Create public and private distribution groups

After completing this module, students will be able to:
- Implement a mobile DevOps strategy

AZ-400T02-A: Implementing Continuous Integration

Module 1: Implementing Continuous Integration in an Azure DevOps Pipeline
In this module, you’ll be introduced to continuous integration principles including: benefits, challenges, build best practices, and implementation steps. You will also learn about implementing a build strategy with workflows, triggers, agents, and tools.

Lessons
- Continuous Integration Overview
- Implementing a Build Strategy

Lab: Enabling Continuous Integration with Azure Pipelines
Lab: Creating a Jenkins Build Job and Triggering CI

After completing this module, students will:
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps

Module 2: Managing Code Quality and Security Policies
In this module, you will be learn how to manage code quality including: technical debt, SonarCloud, and other tooling solutions. You will also learn how to manage security policies with open source, OWASP, and WhiteSource Bolt.
Lessons
- Managing Code Quality
- Managing Security Policies

Lab: Managing Technical Debt with Azure DevOps and SonarCloud
Lab: Checking Vulnerabilities using WhiteSource Bolt and Azure DevOps

After completing this module, students will be able to:
- Manage code quality including: technical debt, SonarCloud, and other tooling solutions.
- Manage security policies with open source, OWASP, and WhiteSource Bolt.
- Manage code quality including: technical debt, SonarCloud, and other tooling solutions.

Module 3: Implementing a Container Build Strategy
In this module, you will learn how to implement a container strategy including how containers are different from virtual machines and how microservices use containers. You will also learn how to implement containers using Docker.

Lessons
- Implementing a Container Build Strategy

Lab: Existing .NET Applications with Azure and Docker Images

After completing this module, students will be able to:
- Implement a container strategy including how containers are different from virtual machines and how microservices use containers.
- Implement containers using Docker.

AZ-400T03-A: Implementing Continuous Delivery

Module 1: Design a Release Strategy
Lessons
- Introduction to Continuous Delivery
- Release strategy recommendations
- Building a High Quality Release pipeline
- Choosing a deployment pattern
- Choosing the right release management tool

Lab: Building a release strategy

After completing this module, students will be able to:
- Differentiate between a release and a deployment
- Define the components of a release pipeline
- Explain things to consider when designing your release strategy
• Classify a release versus a release process, and outline how to control the quality of both
• Describe the principle of release gates and how to deal with release notes and documentation
• Explain deployment patterns, both in the traditional sense and in the modern sense
• Choose a release management tool

Module 2: Set up a Release Management Workflow

Lessons
• Create a Release Pipeline
• Provision and Configure Environments
• Manage And Modularize Tasks and Templates
• Integrate Secrets with the release pipeline
• Configure Automated Integration and Functional Test Automation
• Automate Inspection of Health

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

Lab: Setting up secrets in the pipeline with Azure Key vault

Lab: Setting up and Running Load Tests

Lab: Setting up and Running Functional Tests

Lab: Using Azure Monitor as release gate

Lab: Creating a Release Dashboard

After completing this module, students will 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
• Classify an Agent, Agent Queue and Agent Pool
• Explain why you sometimes need multiple release jobs in one release pipeline
• Differentiate between multi-agent and multi-configuration release job
• Use release variables and stage variables in your release pipeline
• Deploy to an environment securely, using a service connection
• Embed testing in the pipeline
• List the different ways to inspect the health of your pipeline and release by using, alerts, service hooks and reports
• Create a release gate
Module 3: Implement an appropriate deployment pattern

Lessons
- Introduction into Deployment Patterns
- Implement Blue Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching
- AB Testing
- Progressive Exposure Deployment

Lab: Blue-Green Deployments

Lab: Traffic Manager

After completing this module, students will be able to:
- Describe deployment patterns
- Implement Blue Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment

AZ-400T04-A: Implementing Dependency Management

Module 1: Designing a Dependency Management Strategy

Lessons
- Introduction
- Packaging dependencies
- Package management
- Implement a versioning strategy

Lab: Updating packages

After completing this module, students will be able to:
- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance
Module 2: Manage security and compliance

Lessons
- Introduction
- Package security
- Open source software
- Integrating license and vulnerability scans

After completing this module, students will be able to:
- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds

AZ-400T05-A: Implementing Application Infrastructure

Module 1: Infrastructure and Configuration Azure Tools

Lessons
- Learning Objectives
- Infrastructure as Code and Configuration Management
- Create Azure Resources using ARM Templates
- Create Azure Resources using Azure CLI
- Create Azure Resources by using Azure PowerShell
- Additional Automation Tools
- Version Control
- Lab Deploy to Azure using ARM templates
- Module Review Questions

After completing this module, students will be able to:
- Apply infrastructure and configuration as code principles
- Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI

Module 2: Azure Deployment Models and Services

Lessons
- Learning Objectives
- Deployment Models and Options
- Azure Infrastructure-as-a-Service (IaaS) Services
- Azure Automation with DevOps
- Desired State Configuration (DSC)
- Azure Platform-as-a-Service (PaaS) services
- Azure Service Fabric
- Lab Azure Automation - IaaS or PaaS deployment
- Module Review Questions

After completing this module, students will be able to:
- Describe deployment models and services that are available with Azure
Module 3: Create and Manage Kubernetes Service Infrastructure
Lessons
- Learning Objectives
- Azure Kubernetes Service
- Lab Deploy and Scale AKS Cluster
- Module Review Questions
After completing this module, students will be able to:
- Deploy and configure a Managed Kubernetes cluster

Module 4: Third Party and Open Source Tools available with Azure
Lessons
- Learning Objectives
- Chef
- Puppet
- Ansible
- Cloud-Init
- Terraform
- Lab Provision and configure an App in Azure Using X
- Module Review Questions
After completing this module, students will be able to:
- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform

Module 5: Implement Compliance and Security in your Infrastructure
Lessons
- Security and Compliance Principles with DevOps
- Azure Security Center
- Lab Integrate a scanning extension or tool in an AZ DevOps pipeline/security center
- Module Review Questions
After completing this module, students will be able to:
- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure

Module 6: Course Completion
Lessons
- Final Exam

AZ-400T06-A: Implementing Continuous Feedback
Module 1: Recommend and design system feedback mechanisms
Lessons
- The inner loop
- Continuous Experimentation mindset
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback
• Design process to automate application analytics

Lab: Integration between Azure DevOps and Teams

Lab: Feature Flags

After completing this module, students will be able to:
• Design practices to measure end-user satisfaction
• Design processes to capture and analyze user feedback from external sources
• Design routing for client application crash report data
• Recommend monitoring tools and technologies
• Recommend system and feature usage tracking tools

Module 2: Implement process for routing system feedback to development teams

Lessons
• 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

After completing this module, students will be able to:
• Configure crash report integration for client applications
• Develop monitoring and status dashboards
• Implement routing for client application crash report data
• Implement tools to track system usage, feature usage, and flow
• Integrate and configure ticketing systems with development team's work management

Module 3: Optimize feedback mechanisms

Lessons
• Site Reliability Engineering
• Analyze telemetry to establish a baseline
• Perform ongoing tuning to reduce meaningless or non-actionable alerts
• Analyze alerts to establish a baseline
• Blameless PostMortems and a Just Culture

After completing this module, students will be able to:
• Analyze alerts to establish a baseline
• Analyze telemetry to establish a baseline
• Perform live site reviews and capture feedback for system outages
• Perform ongoing tuning to reduce meaningless or non-actionable alerts
AZ-400T07-A: Designing a DevOps Strategy

Module 1: Planning for DevOps
In this module, students will learn about transformation planning, project selection, and team structures.

Lessons
- Transformation Planning
- Project Selection
- Team Structures

Lab: Agile Planning and Portfolio Management with Azure Boards
After completing this module students will be able to:
- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and KPIs
- Create a team and agile organizational structure

Module 2: Planning for Quality and Security
In this module, students will learn about developing a quality strategy and planning for secure development.

Lessons
- Planning a Quality Strategy
- Planning Secure Development

Lab: Feature Flag Management with LaunchDarkly and AzureDevOps
After completing the module, students will be able to:
- Develop a project quality strategy
- Plan for secure development practices and compliance rules.

Module 3: Migrating and Consolidating Artifacts and Tools
In this module, students will learn about migrating and consolidating artifacts, and migrating and integrating source control measures.

Lessons
- Migrating and Consolidating Artifacts
- Migrating and Integrating Source Control

Lab: Integrating Azure Repos and Azure Pipelines with Eclipse
After completing this module, students will be able to:
- Migrate and consolidate artifacts
- Migrate and integrate source control measures