

Designing and Implementing Microsoft DevOps Solutions

Course AZ-400

4 Days

Instructor-led, Hands-on

Introduction

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.

Who Should Attend?

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

At Course Completion

After completing this course, students will be able to:

- Plan Agile with GitHub Projects and Azure Boards
- Design and implement branch strategies and workflows
- Collaborate with pull requests in Azure Repos
- Explore Git hooks
- Plan foster inner source
- Manage and configure repositories
- Identify technical debt
- Explore Azure Pipelines
- Manage Azure Pipeline agents and pools
- Describe pipelines and concurrency
- Design and implement a pipeline strategy
- Integrate with Azure Pipelines
- Introduction to GitHub Actions
- Learn continuous integration with GitHub Actions
- Design a container build strategy
- Create a release pipeline
- Explore release recommendations
- Provision and test environments
- Manage and modularize tasks and templates
- Automate inspection of health
- Introduction to deployment patterns

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Implement blue-green deployment and feature toggles
- Implement canary releases and dark launching
- Implement A/B testing and progressive exposure deployment
- Integrate with identity management systems
- Manage application configuration data
- 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)
- Implement Bicep
- Introduction to Secure DevOps
- Implement open-source software
- Software Composition Analysis
- Security Monitoring and Governance
- Explore package dependencies
- Understand package management
- Migrate consolidate and secure artifacts
- Implement a versioning strategy
- Introduction to GitHub Packages
- Implement tools to track usage and flow
- Develop monitor and status dashboards
- Share knowledge within teams
- Design processes to automate application analytics
- Manage alerts, blameless retrospectives and a just culture

Prerequisites

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.

Course Materials

The student kit includes a workbook and other necessary materials for this class.

Course Outline

Module 1: Introduction to DevOps

- What is DevOps?
- Explore the DevOps journey
- Identify transformation teams
- Define organization structure for agile practices
- Explore shared goals and define timelines
- What is Azure DevOps?
- What is GitHub?
- Design a license management strategy
- What is source control?
- Describe working with Git locally

Module 2: Plan Agile with GitHub Projects and Azure Boards

- Configure projects and teams in Azure DevOps
- Link GitHub to Azure Boards
- Configure GitHub Projects
- Manage work with GitHub Project boards
- Customize Project views
- Collaborate using team discussions
- Design and implement a strategy for feedback cycles
- Design and implement source, bug, and quality traceability
- Agile Plan and Portfolio Management with Azure Boards

Module 3: Design and implement branch strategies and workflows

- Explore branch workflow types
- Explore feature branch workflow
- Explore Git branch model for continuous delivery
- Explore GitHub flow
- Explore fork workflow
- Implement branch merging restrictions
- Version Control with Git in Azure Repos

Module 4: Collaborate with pull requests in Azure Repos

- Collaborate with pull requests

Module 5: Explore Git hooks

- Implement Git hooks

Module 6: Plan foster inner source

- Explore foster inner source
- Implement the fork workflow
- Describe inner source with forks

Module 7: Manage and configure repositories

- Work with large repositories
- Explore monorepo versus multiple repos
- Implement a change log
- Implement Scalar and cross repo when cross repo sharing
- Recover specific data by using Git commands
- Purge repository data
- Manage releases with GitHub Repos
- Automate release notes with GitHub
- Create API documentation
- Implement automation of Git history documentation
- Configure source control repository permissions using GitHub
- Configure GitHub tags to organize repositories

Module 8: Identify technical debt

- Examine code quality
- Examine complexity and quality metrics
- Measure and manage technical debt
- Integrate other code quality tools
- Plan effective code reviews

Module 9: Explore Azure Pipelines

- Explore the concept of pipelines in DevOps
- Describe Azure Pipelines
- Understand Azure Pipelines key terms

Module 10: Manage Azure Pipeline agents and pools

- Choose between Microsoft-hosted versus self-hosted agents
- Explore job types
- Explore predefined agent pool
- Understand typical situations for agent pools
- Communicate with Azure Pipelines
- Communicate to deploy to target servers
- Examine other considerations
- Describe security of agent pools
- Configure agent pools and understand pipeline styles

Module 11: Describe pipelines and concurrency

- Understand parallel jobs
- Estimate parallel jobs
- Describe Azure Pipelines and open-source projects
- Explore Azure Pipelines and Visual Designer
- Describe Azure Pipelines and YAML
- Enable Continuous Integration with Azure Pipelines

Module 12: Design and implement a pipeline strategy

- Configure agent demands
- Explore multi-configuration and multi-agent
- Integrate GitHub repos and Azure Pipelines
- Design and implement a comprehensive testing strategy
- Implement code coverage and show in the pipeline
- Implement multi-job builds
- Explore source control types supported by Azure Pipelines

Module 13: Integrate with Azure Pipelines

- Describe the anatomy of a pipeline
- Understand the pipeline structure
- Detail templates
- Explore YAML resources
- Use multiple repositories in your pipeline
- Migrate a pipeline from classic to YAML in Azure Pipelines

Module 14: Introduction to GitHub Actions

- What are actions?
- Explore Actions flow
- Understand Workflows
- Describe standard workflow syntax elements
- Explore Events
- Explore Jobs
- Explore Runners
- Examine release and test an action

Module 15: Learn continuous integration with GitHub Actions

- Describe continuous integration with actions
- Examine environment variables
- Share artifacts between jobs
- Examine Workflow badges
- Describe best practices for creating actions
- Mark releases with Git tags

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Create encrypted secrets
- Use secrets in a workflow
- Implement GitHub Actions for CI/CD

Module 16: Design a container build strategy

- Examine structure of containers
- Work with Docker containers
- Understand Dockerfile core concepts
- Examine multi-stage dockerfiles
- Examine considerations for multiple stage builds
- Explore Azure container-related services
- Deploy Docker containers to Azure App Service web apps

Module 17: Create a release pipeline

- Describe Azure DevOps release pipeline capabilities
- Explore release pipelines
- Explore artifact sources
- Choose the appropriate artifact source
- Examine considerations for deployment to stages
- Explore build and release tasks
- Explore custom build and release tasks
- Explore release jobs
- Understand database deployment task
- Configure Pipelines as Code with YAML

Module 18: Explore release recommendations

- Understand the delivery cadence and three types of triggers
- Explore release approvals
- Explore release gates
- Use release gates to protect quality
- Explore GitOps release strategy and recommendations
- Control Deployments using Release Gates

Module 19: Provision and test environments

- Provision and configure target environments
- Configure automated integration and functional test automation
- Understand Shift-left
- Set up and run availability tests
- Explore Azure Load Testing
- Set up and run functional tests

Module 20: Manage and modularize tasks and templates

- Examine task groups
- Explore variables in release pipelines

Module 21: Automate inspection of health

- Automate inspection of health
- Explore events and notifications
- Explore service hooks
- Configure Azure DevOps notifications
- Configure GitHub notifications
- Explore how to measure quality of your release process
- Examine release notes and documentation
- Examine considerations for choosing release management tools
- Explore common release management tools

Module 22: Introduction to deployment patterns

- Explore microservices architecture
- Examine classical deployment patterns
- Understand modern deployment patterns

Module 23: Implement blue-green deployment and feature toggles

- What is blue-green deployment?
- Explore deployment slots
- Describe feature toggle maintenance

Module 24: Implement canary releases and dark launching

- Explore canary releases
- Examine traffic manager
- Understand dark launching

Module 25: Implement A/B testing and progressive exposure deployment

- What is A/B testing?
- Explore CI-CD with deployment rings

Module 26: Integrate with identity management systems

Integrate GitHub with single sign-on (SSO)

- Design and implement permissions and roles in GitHub
- Design and implement permissions and security groups in Azure DevOps

- Explore workload identities
- Implement managed identities

Module 27: Manage application configuration data

- Rethink application configuration data
- Explore separation of concerns
- Understand external configuration store patterns
- Implement Azure DevOps secure files
- Examine Key-value pairs
- Examine App configuration feature management
- Integrate Azure Key Vault with Azure Pipelines
- Manage secrets, tokens and certificates
- Examine DevOps inner and outer loop
- Integrate Azure Key Vault with Azure DevOps
- Enable Dynamic Configuration and Feature Flags

Module 28: Explore infrastructure as code and configuration management

- Explore environment deployment
- Examine environment configuration
- Understand imperative versus declarative configuration
- Understand idempotent configuration

Module 29: Create Azure resources using Azure Resource Manager templates

- Why use Azure Resource Manager templates?
- Explore template components
- Manage dependencies
- Modularize templates
- Manage secrets in templates

Module 30: Create Azure resources by using Azure CLI

- What is Azure CLI?
- Work with Azure CLI

Module 31: Explore Azure Automation with DevOps

- Create automation accounts
- What is a runbook?
- Understand automation shared resources
- Explore runbook gallery
- Examine webhooks
- Explore source control integration

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Explore PowerShell workflows
- Create a workflow
- Explore hybrid management
- Examine checkpoint and parallel processing

Module 32: Implement Desired State Configuration (DSC)

- Understand configuration drift
- Explore Desired State Configuration (DSC)
- Explore Azure Automation State configuration (DSC)
- Examine DSC configuration file
- Implement DSC and Linux Automation on Azure

Module 33: Implement Bicep

- What is Bicep?
- Install Bicep
- Understand Bicep file structure and syntax
- Deployments using Azure Bicep templates

Module 34: Introduction to Secure DevOps

- Describe SQL injection attack
- Understand DevSecOps
- Explore Secure DevOps Pipeline
- Explore key validation points
- Explore continuous security validation
- Understand threat modeling
- Explore CodeQL in GitHub

Module 35: Implement open-source software

- Explore how software is built
- What is open-source software?
- Explore corporate concerns with open-source software components
- Explore common open-source licenses
- Examine license implications and ratings

Module 36: Software Composition Analysis

- Inspect and validate code bases for compliance
- Explore software composition analysis (SCA)
- Implement GitHub Dependabot alerts and security updates
- Integrate software composition analysis checks into pipelines
- Examine tools for assess package security and license rate
- Automate container scanning, including container images
- Interpret alerts from scanner tools

Module 37: Security Monitoring and Governance

- Implement pipeline security
- Explore Microsoft Defender for Cloud
- Examine Microsoft Defender for Cloud usage scenarios
- Explore Azure Policy
- Understand policies
- Explore initiatives
- Explore resource locks
- Understand Microsoft Defender for Identity
- Integrate GitHub Advanced Security with Microsoft Defender for Cloud
- Configure GitHub Advanced Security for GitHub and Azure DevOps

Module 38: Explore package dependencies

- What is dependency management?
- Describe elements of a dependency management strategy
- Identify dependencies
- Understand source and package componentization
- Decompose your system
- Scan your codebase for dependencies

Module 39: Understand package management

- Explore packages
- Understand package feeds
- Explore package feed managers
- Explore common public package sources
- Explore self-hosted and SaaS based package sources
- Consume packages
- Publish packages
- Package management with Azure Artifacts

Module 40: Migrate consolidate and secure artifacts

- Identify existing artifact repositories
- Migrate and integrating artifact repositories
- Secure access to package feeds
- Examine roles
- Examine permissions
- Examine authentication

Module 41: Implement a versioning strategy

- Understand versioning of artifacts
- Explore semantic versioning
- Examine release views

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Promote packages
- Explore best practices for versioning

Module 42: Introduction to GitHub Packages

- Publish packages
- Install a package
- Delete and restore a package
- Explore package access control and visibility

Module 43: Implement tools to track usage and flow

- Understand the inner loop
- Explore Azure Monitor and Log Analytics
- Examine Kusto Query Language (KQL)
- Explore Application Insights
- Implement Application Insights
- Design and implement metrics and queries
- Monitor application performance with Application Insights

Module 44: Develop monitor and status dashboards

- Configure monitoring in GitHub
- Explore Azure Dashboards
- Explore Azure Monitor workbooks
- Explore Power BI
- Build your own custom application
- Monitor pipeline health, including failure rate, duration, and flaky tests
- Optimize a pipeline for cost, time, performance, and reliability
- Optimize pipeline concurrency for performance and cost

Module 45: Share knowledge within teams

- Share acquired knowledge within development teams
- Integrate GitHub and Azure DevOps with Microsoft Teams

Share team knowledge using Azure Project Wiki

Module 46: Design processes to automate application analytics

- Explore rapid responses and augmented search
- Integrate telemetry
- Examine monitoring tools and technologies
- Explore IT Service Management Connector

Module 47: Manage alerts, blameless retrospectives and a just culture

- Examine when to get a notification
- Explore how to fix it
- Explore smart detection notifications
- Improve performance
- Understand server response time degradation
- Reduce meaningless and non-actionable alerts
- Examine blameless retrospective
- Develop a just culture