Designing and Implementing Microsoft DevOps Solutions (AZ-400T00)

COURSE OVERVIEW

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 WILL BENEFIT FROM THIS COURSE?

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

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.

If you are new to Azure and cloud computing, consider one of the following resources:

- Instructor-led course: AZ-900: Azure Fundamentals

COURSE OBJECTIVES

After completing this course, students will be able to:

- Introduction to DevOps
- Choose the right project
- Describe team structures
- Choose the DevOps tools
- Plan Agile with GitHub Projects and Azure Boards
- Introduction to source control
- Describe types of source control systems
- Work with Azure Repos and GitHub
- Structure your Git Repo
- Manage Git branches and workflows
- Collaborate with pull requests in Azure Repos

- Identify technical debt
- Explore Git hooks
- Plan foster inner source
- Manage Git repositories
- Explore Azure Pipelines
- Manage Azure Pipeline agents and pools
- Describe pipelines and concurrency
- Explore continuous integration
- Implement a pipeline strategy
- Integrate with Azure Pipelines
- Introduction to GitHub Actions
- Learn continuous integration with GitHub Actions
- Design a container build strategy
- Introduction to continuous delivery
- 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
- 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
- Static analyzers
- OWASP and Dynamic Analyzers
- Security Monitoring and Governance
- Explore package dependencies
- Understand package management
- Migrate consolidating 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

COURSE OUTLINE

Module 1: Introduction to DevOps

- Understand what DevOps is and the steps to accomplish it
- Identify teams to implement the process
- Plan for the transformation with shared goals and timelines
- Plan and define timelines for goals

Module 2: Choose the right project

- Understand different projects and systems to guide the journey
- Select a project to start the DevOps transformation
- Identify groups to minimize initial resistance
- Identify project metrics and Key Performance Indicators (KPI's)

Module 3: Describe team structures

- Understand agile practices and principles of agile development
- Create a team and agile organizational structure
- Identify ideal DevOps team members
- Select and configure tools for collaboration

Module 4: Choose the DevOps tools

- 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

Module 5: Plan Agile with GitHub Projects and Azure Boards

- Describe GitHub Projects and Azure Boards
- Link Azure Boards and GitHub
- Configure and Manage GitHub Projects and boards
- Customize Project views

Module 6: Introduction to source control

- Understand source control
- Apply best practices for source control
- Describe the benefits of using source control

Module 7: Describe types of source control systems

- Apply source control practices in your development process
- Explain the differences between centralized and distributed version control
- Understand Git and TFVC
- Develop using Git

Module 8: Work with Azure Repos and GitHub

- Describe Azure Repos and GitHub
- Migrate from TFVC to Git
- Work with GitHub Codespaces

Module 9: Structure your Git Repo

- Understand Git repositories
- Implement mono repo or multiple repos
- Explain how to structure Git Repos
- Implement a change log

Module 10: Manage Git branches and workflows

- Describe Git branching workflows
- Implement feature branches
- Implement GitHub Flow
- Fork a repo

Module 11: Collaborate with pull requests in Azure Repos

- Leverages pull requests for collaboration and code reviews
- Give feedback using pull requests
- Configure branch policies
- Use GitHub mobile for pull requests approvals

Module 12: Identify technical debt

- Identify and manage technical debt
- Integrate code quality tools
- Plan code reviews
- Describe complexity and quality metrics

Module 13: Explore Git hooks

- Understand Git hooks
- Identify when used Git hooks
- Implement Git hooks for automation
- Explain Git hooks' behavior

Module 14: Plan foster inner source

- Use Git to foster inner sources across the organization
- Implement fork workflow
- Choose between branches and forks
- Share code between forks

Module 15: Manage Git repositories

- Understand large Git repositories
- Explain VFS for Git
- Use Git Large File Storage (LFS)
- Purge repository data
- Manage and Automate Release Notes with GitHub

Module 16: Explore Azure Pipelines

- Describe Azure Pipelines
- Explain the role of Azure Pipelines and its components
- Decide Pipeline automation responsibility
- Understand Azure Pipeline key terms

Module 17: Manage Azure Pipeline agents and pools

- Choose between Microsoft-hosted and self-hosted agents
- Install and configure Azure Pipelines Agents
- Configure agent pools
- Make the agents and pools secure
- Explore communication to deploy using Azure Pipelines

Module 18: Describe pipelines and concurrency

- Use and estimate parallel jobs
- Use Azure Pipelines for open-source or private projects
- Use Visual Designer
- Work with Azure Pipelines and YAML

Module 19: Explore continuous integration

- Explain why continuous integration matters
- Implement continuous integration using Azure Pipelines
- Explain the benefits of continuous integration
- Describe build properties

Module 20: Implement a pipeline strategy

- Define a build strategy
- Explain and configure demands
- Implement multi-agent builds
- Use different source control types available in Azure Pipelines

Module 21: Integrate with Azure Pipelines

- Describe advanced Azure Pipelines anatomy and structure
- Detail templates and YAML resources
- Implement and use multiple repositories

Module 22: Introduction to GitHub Actions

- Explain GitHub Actions and workflows
- Create and work with GitHub Actions and Workflows
- Describe Events, Jobs, and Runners
- Examine output and release management for actions

Module 23: Learn continuous integration with GitHub Actions

- Implement Continuous Integration with GitHub Actions
- Use environment variables
- Share artifacts between jobs and use Git tags
- Create and manage secrets

Module 24: Design a container build strategy

- Design a container strategy
- Work with Docker Containers
- Create an Azure Container Registry
- Explain Docker microservices and containers

Module 25: Introduction to continuous delivery

- Explain continuous delivery (CD)
- Implement continuous delivery in your development cycle
- Understand releases and deployment
- Identify project opportunities to apply CD

Module 26: Create a release pipeline

- 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
- Implement release jobs

Module 27: Explore release recommendations

- Explain things to consider when designing your release strategy
- Define the components of a release pipeline and use artifact sources
- Create a release approval plan
- Implement release gates

Module 28: Provision and test environments

- Provision and configure target environment
- Deploy to an environment securely using a service connection
- Configure functional test automation and run availability tests
- Setup test infrastructure

Module 29: Manage and modularize tasks and templates

- Use and manage task and variable groups
- Use release variables and stage variables in your release pipeline
- Use variables in release pipelines

Module 30: Automate inspection of health

- Implement automated inspection of health
- Create and configure events
- Configure notifications in Azure DevOps and GitHub
- Create service hooks to monitor the pipeline
- Classify a release versus a release process, and outline how to control the quality of both
- Choose a release management tool

Module 31: Introduction to Deployment Patterns

- Describe deployment patterns
- Explain microservices architecture
- Understand classical and modern deployment patterns
- Plan and design your architecture

Module 32: Implement blue-green deployment and feature toggles

- Explain deployment strategies
- Implement blue green deployment
- Understand deployment slots
- Implement and manage feature toggles"

Module 33: Implement canary releases and dark launching

- Describe deployment strategies
- Implement canary release
- Explain traffic manager
- Understand dark launching

Module 34: Implement A/B testing and progressive exposure deployment

- Implement progressive exposure deployment
- Implement A/B testing
- Implement CI/CD with deployment rings
- Identify the best deployment strategy

Module 35: Integrate with identity management systems

- Integrate Azure DevOps with identity management systems
- Integrate GitHub with single sign-on (SSO)
- Understand and create a service principal
- Create managed service identities

Module 36: Manage application configuration data

- Rethink application configuration data
- Understand separation of concerns
- Integrate Azure Key Vault with Azure Pipelines
- Manage secrets, tokens, and certificates
- Describe Azure App Configuration
- Understand Key-value pairs
- Understand app configuration feature management
- Implement application configuration

Module 37: Explore infrastructure as code and configuration management

- Understand how to deploy your environment
- Plan your environment configuration
- Choose between imperative versus declarative configuration
- Explain idempotent configuration

Module 38: Create Azure resources using Azure Resource Manager templates

- Create Azure resources using Azure Resource Manager templates
- Understand Azure Resource Manager templates and template components
- Manage dependencies and secrets in templates
- Organize and modularize templates

Module 39: Create Azure resources by using Azure CLI

- Create Azure resources using Azure CLI
- Understand and work with Azure CLI
- Run templates using Azure CLI
- Explains Azure CLI commands

Module 40: Explore Azure Automation with DevOps

- Implement automation with Azure DevOps
- Create and manage runbooks
- Create webhooks
- Create and run a workflow runbook and PowerShell workflows

Module 41: Implement Desired State Configuration (DSC)

- Implement Desired State Configuration (DSC)
- Describe Azure Automation State Configuration
- Implement DSC and Linux Automation on Azure
- Plan for hybrid management

Module 42: Implement Bicep

- Learn what Bicep is
- Learn how to install it and create a smooth authoring experience
- Use Bicep to deploy resources to Azure
- Deploy Bicep files in Cloud Shell and Visual Studio Code

Module 43: Introduction to Secure DevOps

- Identify SQL injection attack
- Understand DevSecOps
- Implement pipeline security
- Understand threat modeling

Module 44: Implement open-source software

- Implement open-source software
- Explain corporate concerns for open-source components
- Describe open-source licenses
- Understand the license implications and ratings

Module 45: Software Composition Analysis

- Inspect and validate code bases for compliance
- Integrate security tools like WhiteSource with Azure DevOps
- Implement pipeline security validation
- Interpret alerts from scanning tools
- Configure GitHub Dependabot alerts and security

Module 46: Static analyzers

- Understand Static Analyzers
- Work with SonarCloud
- Work with CodeQL in GitHub
- Interpret alerts from scanning tools

Module 47: OWASP and Dynamic Analyzers

- Understand OWASP and Dynamic Analyzers
- Implement OWASP Security Coding Practices
- Understand compliance for code bases

Module 48: Security Monitoring and Governance

- Configure Microsoft Defender for Cloud
- Understand Azure policies
- Describe initiatives, resource locks, and Azure Blueprints
- Work with Microsoft Defender for Identity

Module 49: Explore package dependencies

- Define dependency management strategy
- Identify dependencies
- Describe elements and componentization of a dependency management
- Scan your codebase for dependencies

Module 50: Understand package management

- Implement package management
- Manage package feed
- Consume and create packages
- Publish packages

Module 51: Migrate consolidate and secure artifacts

- Identify artifact repositories
- Migrate and integrate artifact repositories
- Secure package feeds
- Understand roles, permissions, and authentication

Module 52: Implement a versioning strategy

- Implement a versioning strategy
- Promote packages
- Push packages from pipeline
- Describe semantic and explore best practices for versioning

Module 53: Introduction to GitHub Packages

- Publish packages
- Install packages
- Delete and restore packages
- Configure access control and visibility

Module 54: Implement tools to track usage and flow

- Implement tools to track feedback
- Plan for continuous monitoring
- Implement Application Insights
- Use Kusto Query Language (KQL)

Module 55: Develop monitor and status dashboards

- Configure Azure Dashboards
- Work with View Designer in Azure Monitor
- Create Azure Monitor Workbooks
- Monitor with Power BI

Module 56: Share knowledge within teams

- Share knowledge with development teams
- Work with Azure DevOps Wikis
- Integrate with Azure Boards

Module 57: Design processes to automate application analytics

- Automate application analytics
- Assist DevOps with rapid responses and augmented search
- Integrate telemetry
- Implement monitoring tools and technologies

Module 58: Manage alerts, blameless retrospectives, and a just culture

- Carry out blameless retrospectives and create a just culture
- Improve application performance
- Explain server response time degradation
- Reduce meaningless and non-actionable alerts

WHY TRAIN WITH SUNSET LEARNING INSTITUTE?

Sunset Learning Institute (SLI) has been an innovative leader in developing and delivering authorized technical training since 1996. Our goal is to help our customers optimize their technology Investments by providing convenient, high quality technical training that our customers can rely on. We empower students to master their desired technologies for their unique environments.

What sets SLI apart is not only our immense selection of trainings options, but our convenient and consistent delivery system. No matter how complex your environment is or where you are located, SLI is sure to have a training solution that you can count on!

Premiere World Class Instruction Team

- All SLI instructors have a four-year technical degree, instructor level certifications and field consulting work experience
- Sunset Learning has won numerous Instructor Excellence and Instructor Quality Distinction awards since 2012

Enhanced Learning Experience

- The goal of our instructors during class is ensure students understand the material, guide them through our labs and encourage questions and interactive discussions.

Convenient and Reliable Training Experience

- You have the option to attend classes live with the instructor, at any of our established training facilities, or from the convenience of your home or office
- All Sunset Learning Institute classes are guaranteed to run – you can count on us to deliver the training you need when you need it!

Outstanding Customer Service

- You will work with a dedicated account manager to suggest the optimal learning path for you and/or your team
- An enthusiastic student services team is available to answer any questions and ensure a quality training experience

Interested in Private Group Training?

Contact Us