AZ-400T00 Design and Implement Microsoft DevOps solutions

Category: Cloud Platforms | **Vendor:** Microsoft Technical

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format:	Instructor Led - Online
Certification:	Microsoft Certified: DevOps Engineer Expert
Related Exam:	AZ-400

Course Overview

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

About This Course

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.

Prerequisites & Entry Requirements

General 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.
- AZ-104T00 - Microsoft Azure Administrator
- AZ-204T00: Developing Solutions for Microsoft Azure

Learning Outcomes

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

Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure

PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises

and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment

pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce

meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git

reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational

structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application

infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure

PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises

and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application analyticsManage alerts and reduce meaningless and non-actionable alertsCarry out blameless retrospectives and create a just cultureDefine an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructureImplement compliance and security in your application infrastructureDescribe the potential challenges with integrating open-source softwareInspect open-source software packages for security and license complianceManage organizational security and compliance policiesIntegrate license and vulnerability scans into build and deployment pipelinesConfigure build pipelines to access package security and license ratings, Course ObjectivesPlan for the transformation with shared goals and timelinesSelect a project and identify project metrics and Key Performance Indicators (KPI's)Create a team and agile organizational structureDesign a tool integration strategyDesign a license management strategy (e.g., Azure DevOps and GitHub users)Design a strategy for end-to-end traceability from work items to working softwareDesign an authentication and access strategyDesign a strategy for integrating on-premises and cloud resourcesDescribe the benefits of using Source ControlDescribe Azure Repos and GitHubMigrate from TFVC to GitManage code quality including technical debt SonarCloud, and other tooling solutionsBuild organizational knowledge on code qualityExplain how to structure Git reposDescribe Git branching workflowsLeverage pull requests for collaboration and code reviewsLeverage Git hooks for automationUse Git to foster inner source across the organizationExplain the role of Azure Pipelines and its componentsConfigure Agents for use in Azure PipelinesExplain why continuous integration mattersImplement continuous integration using Azure PipelinesDefine Site Reliability EngineeringDesign processes to measure end-user satisfaction and analyze user feedbackDesign processes to automate application ana

Detailed Course Outline

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 assessment

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 assessment

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 assessment

4 - Collaborate with pull requests in Azure Repos

- Collaborate with pull requests
- Module assessment

5 - Explore Git hooks

- Implement Git hooks
- Module assessment

6 - Plan to foster inner source

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

7 - Manage and configure repositories

- Work with large repositories
- Explore monorepo versus multiple repos
- Implement a changelog
- Implement Scalar and cross-repository sharing
- Recover specific data by using Git commands
- Clean up repository data
- Manage releases with GitHub repos
- Automate release notes with GitHub
- Create API documentation
- Implement automation of Git history documentation
- Configure repository permissions using GitHub
- Configure GitHub tags to organize repositories

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

9 - Explore Azure Pipelines

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

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 assessment

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 assessment

12 - Design and implement a pipeline strategy

- Configure agent demands
- Explore multi-configuration and multi-agent builds
- Integrate GitHub repositories with 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

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 assessment

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 assessment

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

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 assessment

17 - Create a Release Pipeline

- Describe Azure Pipelines 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 assessment

18 - Explore release strategy 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 assessment

19 - Configure and Provision 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 assessment

20 - Manage and Modularize Tasks and Templates

- Examine task groups
- Explore Variables in release pipelines
- Module assessment

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 assessment

22 - Introduction to deployment patterns

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

23 - Implement blue-green deployment and feature toggles

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

24 - Implement canary releases and dark launching

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

25 - Implement A/B testing and progressive exposure deployment

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

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 assessment

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 Pipelines
- Enable dynamic configuration and feature flags
- Module assessment

28 - Explore Infrastructure as Code and configuration management

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

29 - Create Azure Resources by using Azure Resource Manager templates

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

30 - Create Azure Resources Using Azure CLI

- What Is Azure CLI?
- Install Azure CLI
- Execute Azure CLI Commands Interactively
- Create an Azure CLI Script for Bash
- Tips to Use the Azure CLI Successfully
- Module assessment

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

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 assessment

33 - Implement Bicep

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

34 - Introduction to Secure DevOps

- Describe SQL injection attack
- Understand DevSecOps
- Explore the secure DevOps pipeline
- Explore key validation points
- Explore continuous security validation
- Understand threat modeling
- Explore CodeQL in GitHub
- Module assessment

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 assessment

36 - Software Composition Analysis

- Inspect and validate code bases for compliance
- Explore software composition analysis
- Implement GitHub Dependabot alerts and security updates
- Integrate software composition analysis checks into pipelines
- Examine tools for assessing package security and license compliance
- Automate container image scanning
- Interpret alerts from scanner tools
- Module assessment

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 assessment

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 assessment

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 assessment

40 - Migrate, consolidate, and secure artifacts

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

41 - Implement a versioning strategy

- Understand versioning of artifacts
- Explore semantic versioning
- Examine release views
- Promote packages
- Explore best practices for versioning
- Module assessment

42 - Introduction to GitHub Packages

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

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 Azure Load Testing
- Module assessment

44 - Develop monitor and status dashboards

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

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 assessment

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 assessment

47 - Manage alerts, blameless retrospectives and a just culture

- Examine when to get a notification
- Explore how to fix performance issues
- Configure smart detection notifications
- Improve application performance
- Understand server response time degradation
- Reduce meaningless and non-actionable alerts
- Examine blameless retrospectives
- Develop a just culture
- Module assessment

, Introduction to DevOps, Choose the right project, Choose the DevOps tools, Describe team structures, Plan Agile with GitHub Projects and Azure Boards, Introduction to source control, Explore Git hooks, Plan foster inner source, Explore Azure Pipelines, Manage Azure Pipeline agents and pools, Manage Git repositories, Explore continuous integration, Describe pipelines and concurrency, Implement a pipeline strategy, Integrate with Azure Pipelines, Learn continuous integration with GitHub Actions, Introduction to GitHub Actions, Provision and test environments, Manage and modularize tasks and templates, Automate inspection of health, Implement blue-green deployment and feature toggles, Implement canary releases and dark launching, Implement A/B testing and progressive exposure deployment, Introduction to deployment patterns, Design a container build strategy, Introduction to continuous delivery, Explore release recommendations, Create a release pipeline, Manage Git branches and workflows, Identify technical debt, Collaborate with pull requests in Azure Repos, Describe types of source control systems, Structure your Git Repo, Work with Azure Repos and GitHub, Create Azure resources using Azure Resource Manager templates, Explore infrastructure as code and configuration management, Manage application configuration data, Integrate with identity management systems, Implement Desired State Configuration (DSC), Explore Azure Automation with DevOps, Create Azure resources by using Azure CLI, Introduction to Secure DevOps, Implement Bicep, Explore package dependencies, Security Monitoring and Governance, Understand package management, Implement a versioning strategy, Migrate consolidating and secure artifacts, OWASP and Dynamic Analyzers, Static analyzers, Implement open-source software, Software Composition Analysis, Develop monitor and status dashboards, Share knowledge within teams, Design processes to automate application analytics, Implement tools to track usage and flow,

Introduction to GitHub Packages, Manage alerts, blameless retrospectives, and a just culture, blameless retrospectives and a just culture, LAB Outline, Development for enterprise DevOps, Implement CI with Azure Pipelines and GitHub Actions, Design and implement a release strategy, Implement a secure continuous deployment using Azure Pipelines, Manage infrastructure as code using Azure and DSC, Implement security and validate code bases for compliance, Design and implement a dependency management strategy, Implement continuous feedback

🎓 Certification Path

Microsoft Certified: DevOps Engineer Expert

Exam: AZ-400

Upcoming Schedule

Available training dates for AZ-400T00 Design and Implement Microsoft DevOps solutions:

Start Date	End Date	Location	Delivery	Price
21 Dec 2026	24 Dec 2026	Online	Virtual	€2,495.00

Frequently Asked Questions

Q: What delivery options are available for AZ-400T00 Design and Implement Microsoft DevOps solutions?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: What certification does this course prepare me for?

The AZ-400T00 Design and Implement Microsoft DevOps solutions course helps prepare you for the Microsoft Certified: DevOps Engineer Expert certification path.

Q: Which exam does this course prepare me for?

This course prepares you for the AZ-400 official exam. You can take this exam at any exam center across United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online wherever you are located.

Q: How many CPD hours does this course provide?

The 4-day AZ-400T00 Design and Implement Microsoft DevOps solutions course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the AZ-400T00 Design and Implement Microsoft DevOps solutions training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for AZ-400T00 Design and Implement Microsoft DevOps solutions?

Yes, we provide corporate training, dedicated training, and closed classes for AZ-400T00 Design and Implement Microsoft DevOps solutions. Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: AZ-400T00: Design and Implement Microsoft DevOps solutions, Microsoft, vILT, AZ-400T00

Q: Why choose Nexus Human for AZ-400T00 Design and Implement Microsoft DevOps solutions?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your AZ-400T00 Design and Implement Microsoft DevOps solutions training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 19/10/2026.