

AZ 400- Design and Implement Microsoft DevOps Solutions



Duration: 4 days

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

Prerequisites: Before attending this course, students must have:

- 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.
- Foundational DevOps concepts, including version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software

Description: The Implementing Cisco Catalyst 9000 Series Switches (ENC9K) v4.0 training provides you with insight into Cisco Catalyst 9000 Series Switches and its solution components. You will learn about the architecture, capabilities, positioning, and implementation of the Cisco Catalyst 9000 Series switches, as well as the management of the switches using CLI, Cisco DNA Center, and IOS-XE web user GUI. Additionally, you will be introduced to security, cloud, automation, and other important features of Cisco Catalyst 9000 Series switches. This course qualifies for 18 Cisco Continuing Education credits towards recertification.

OUTLINE:

LEARNING PATH 1: IMPLEMENT DEVELOPMENT FOR ENTERPRISE DEVOPS

- This learning path examines the key Git features that organizations must plan for when designing their enterprise DevOps.

MODULES

- Introduction to DevOps.
- 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.

LABS:

- Agile planning and portfolio management with Azure Boards
- Version controlling with Git in Azure Repos

LEARNING PATH 2: IMPLEMENT CI WITH AZURE PIPELINES AND GITHUB ACTIONS

- This learning path introduces the continuous integration concept using Azure Pipelines and GitHub Actions and provides instruction on configuring those services and building applications.

MODULES

- 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.

LAB:

- Configuring Agent Pools and Understanding Pipeline Styles
- Enabling continuous integration with Azure Pipelines
- Implementing GitHub Actions for CI/CD
- Deploying Docker containers to Azure App Service web apps

AZ 400- Design and Implement Microsoft DevOps Solutions



LEARNING PATH 3: DESIGN AND IMPLEMENT A RELEASE STRATEGY

- This learning path explains the concepts of continuous delivery and releases strategy considerations for setting up deployment stages and your delivery and deployment cadence, and lastly, setting up your release approvals.

MODULES

- Create a release pipeline.
- Explore release recommendations.
- Provision and test environments.
- Manage and modularize tasks and templates.
- Automate inspection of health.

LABS:

- Configuring pipelines as code with YAML
- Controlling deployments using Release Gates
- Setting up and running functional tests

LEARNING PATH 4: IMPLEMENT A SECURE CONTINUOUS DEPLOYMENT USING AZURE PIPELINES

- This learning path helps you understand automated release gates, secrets, and secret management in your pipeline. You learn how to implement alerting mechanisms, report on your quality, and get notified by using service hooks.

MODULES

- 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.

LABS:

- Integrating Azure Key Vault with Azure DevOps
- Enable Dynamic Configuration and Feature Flags

LEARNING PATH 5: MANAGE INFRASTRUCTURE AS CODE USING AZURE AND DSC

- This learning path explores the “Infrastructure as Code” (IaC) concept and how to manage your operations environment the same way you do applications or other code for general release.

MODULES

- 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.

LAB:

- Deployments using Azure Bicep templates

LEARNING PATH 6: IMPLEMENT SECURITY AND VALIDATE CODE BASES FOR COMPLIANCE

- This learning path explores an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure. It explains compliance and security implementation in your application infrastructure.

MODULES

- Introduction to Secure DevOps.
- Implement open-source software.
- Software Composition Analysis.
- Security Monitoring and Governance.

LEARNING PATH 7: DESIGN AND IMPLEMENT A DEPENDENCY MANAGEMENT STRATEGY

- This learning path examines dependency management in software development, how to identify them in your codebase and how to package and manage dependencies in package feeds.

MODULES

- Explore package dependencies.
- Understand package management.

AZ 400- Design and Implement Microsoft DevOps Solutions



- Migrate, consolidating and secure artifacts.
- Implement a versioning strategy.
- Introduction to GitHub Packages.

LAB:

- Package management with Azure Artifacts

LEARNING PATH 8: IMPLEMENT CONTINUOUS FEEDBACK

- This learning path introduces the continuous feedback concept and describes how to implement it in your DevOps cycle.

MODULES

- 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.

LAB:

- Monitoring Application Performance with Azure Load Testing
- Sharing team knowledge using Azure project wikis