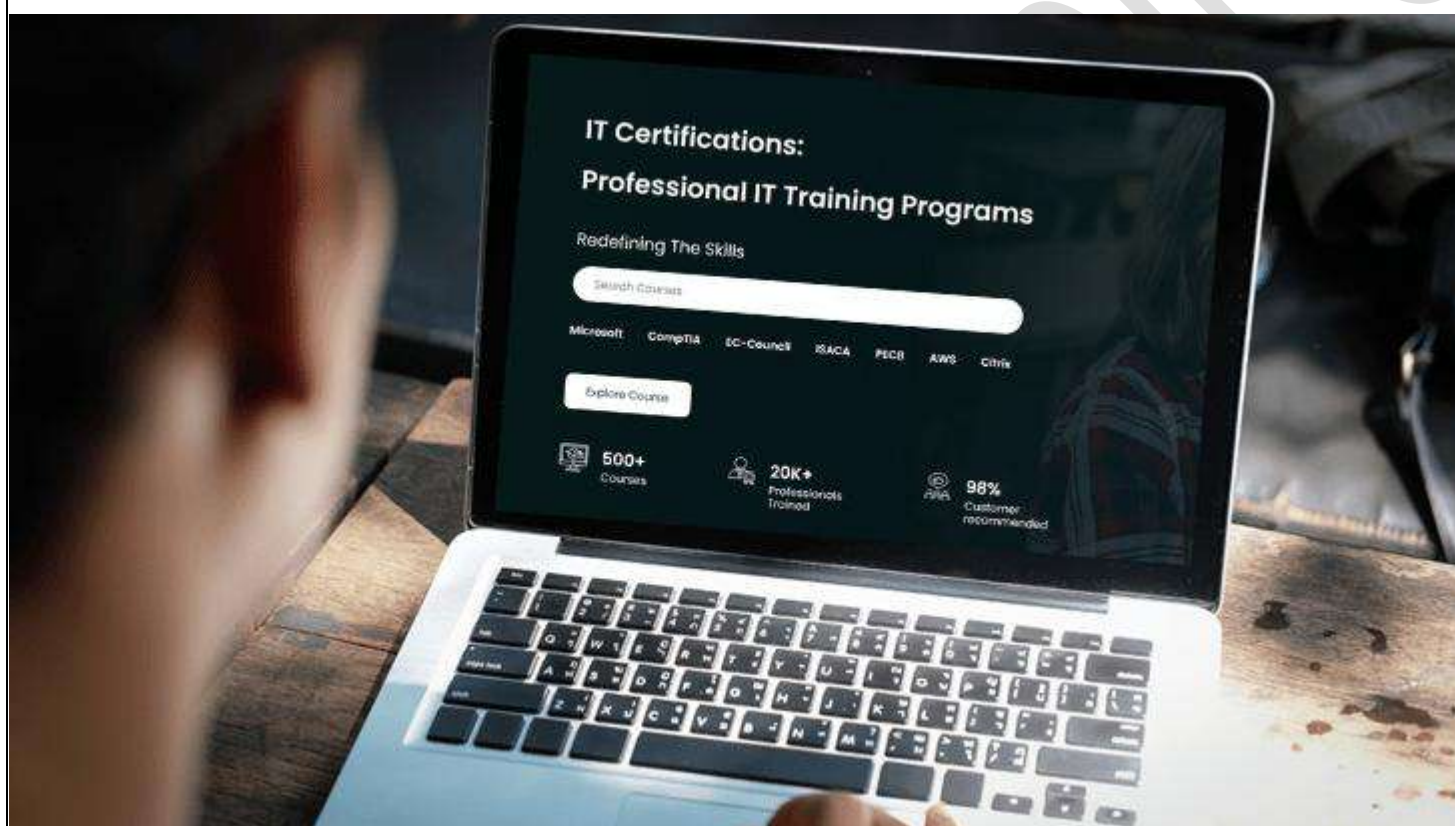




Redefining The Skills



AZ-400T00: DESIGNING AND IMPLEMENTING MICROSOFT DEVOPS SOLUTIONS TRAINING

Duration: 4 Days

Course Description

AZ-400T00: Microsoft Azure DevOps Solution Training is aimed at professionals willing to upskill their understanding for designing a DevOps strategy. This course helps professionals enhance their expertise in DevOps practice while utilizing Microsoft technologies.

While combining the skills of technology and procedures, this course focuses on delivering tailored consumer needs and satisfying business objectives.

We have the Enterprise Training Program which is perfect for organizations and companies. In this framework, candidates will learn the development of DevOps software and implementations of DevOps software building processes.

This training program is perfect for IT professionals who want to transform themselves into Azure DevOps Engineer Expert. It provides in-depth and comprehensive information, which makes it ideal for individuals to prepare for Microsoft's

AZ-400 examination.

This training is designed based on the objectives of the course variant AZ-400T00-A.

Who should attend this course?

- Professionals who are experienced with Azure Developer Associate Certifications or Azure Administrator Associate (AZ-103) who are planning to advance their career in DevOps.
- Given below are professionals who can use Azure DevOps Solution Certification to upskill their current positions:
 - IT Managers
 - Technical Project Managers
 - DevOps Engineers
 - Cloud Solutions Architects
 - Quality Assurance Engineers
 - Operations Support Staff
 - Technical Leads working with cross-functional
 - DevOps teams

What you will learn

- Design a dependency management strategy and secure development process
- Learn about consolidation and migration strategy for DevOps tools
- Create and implement an Agile management approach
- Implement mobile DevOps strategy
- Manage application configuration and secrets
- Employ and integrate source control
- Design a version release strategy and control strategy
- Design a quality strategy and tool integration strategy
- Implement and manage code flow and build infrastructure
- Set up a release management workflow
- Design a container build strategy, build strategy, and a deployment pattern
- Manage security policies and code quality
- Design and optimize system feedback mechanisms
- Design a configuration and an infrastructure management strategy
- Implement infrastructure security and compliance

- Manage Azure Kubernetes Service infrastructure
- Manage compliance and security and implement infrastructure as code (IaC)
- Employ process for routing system feedback

Prerequisites

- The pursuing candidate must have a basic understanding of Azure, version control, core software, and development principles.
- One must be familiar with Agile practices, planning, and portfolio management.
- Working experience in a software development environment is highly beneficial.
- Familiarity with core DevOps terminology and principles.

Recommended

- AZ-900: Microsoft Azure fundamentals
- AZ-104: Microsoft Azure Administrator
- AZ-204: Developing Solutions for Microsoft Azure

Curriculum

Module 1: Development for enterprise DevOps

- 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
- 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
- Design and implement branch strategies and workflows
 - Describe Git branching workflows
 - Implement feature branches
 - Implement GitHub Flow
 - Fork a repo
- 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
- Explore Git hooks
 - Understand Git hooks
 - Identify when used Git hooks
 - Implement Git hooks for automation
 - Explain Git hooks' behavior
- Plan foster inner source
 - Use Git to foster inner source across the organization
 - Implement fork workflow
 - Choose between branches and forks
 - Share code between forks

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

Module 2: Implement CI with Azure Pipelines and GitHub Actions

- Explore Azure Pipelines
 - Describe Azure Pipelines
 - Explain the role of Azure Pipelines and its components
 - Decide Pipeline automation responsibility
 - Understand Azure Pipeline key terms
- 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.
- 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.
- Design and 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
- Integrate with Azure Pipelines
 - Describe advanced Azure Pipelines anatomy and structure
 - Detail templates and YAML resources
 - Implement and use multiple repositories
- Introduction to GitHub Actions
 - Explain GitHub Actions and workflows
 - Create and work with GitHub Actions and Work
 - Describe Events, Jobs and Runners
 - Examine output and release management for actions
- 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.
- Design a container build strategy
 - Design a container strategy
 - Work with Docker Containers

- Create an Azure Container Registry
- Explain Docker microservices and containers

Module 3: Design and implement a release strategy

- 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.
- 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.
- 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.
- 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.
- 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 pipeline.
 - Classify a release versus a release process, and outline how to control the quality of both.
 - Choose a release management tool.

Module 4: Implement a secure continuous deployment using Azure Pipelines

- Introduction to deployment patterns
 - Describe deployment patterns
 - Explain microservices architecture
 - Understand classical and modern deployment patterns
 - Plan and design your architecture
- Implement blue-green deployment and feature toggles
 - Explain deployment strategies.
 - Implement blue green deployment.
 - Understand deployment slots.
 - Implement and manage feature toggles.
- Implement canary releases and dark launching
 - Describe deployment strategies.
 - Implement canary release.
 - Explain traffic manager.
 - Understand dark launching.
- 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.
- 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.
- 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 5: Manage infrastructure as code using Azure and DSC

- 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.
- 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.
- 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.
- 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.
 - Plan for hybrid management.
- 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.
- 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.