Design and Implement Microsoft DevOps solutions Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

The training course provides IT professionals with the knowledge to**design, plan, implement, and optimize DevOps processes** using Microsoft Azure technologies.

Throughout this **4-day advanced training**, you'll explore **source control, build automation, CI/CD, container orchestration, security, and feedback optimization** — preparing you for the**Microsoft Certified: DevOps Engineer Expert** certification.

Bilginç IT Academy is an officially **Accredited Training Organisation (ATO)** by **PeopleCert**, the global leader in professional certification and exam delivery.
As an authorized PeopleCert partner, we deliver accredited training and official certification exams for **ITIL®, PRINCE2®, DevOps, and other PeopleCert-certified programs**, ensuring our learners receive globally recognized credentials with the most up-to-date curriculum.

Visit our official PeopleCert partner page here:Bilginç IT Academy – PeopleCert Official Partner

Important Information:
This course is an official program accredited by PeopleCert and is offered**only together with the corresponding certification exam**. The course fee **includes the exam fee**.
Participants can take the certification exam online through PeopleCert's examination system and earn an internationally recognized certificate.
This practice is mandatory to ensure compliance with PeopleCert's quality standards and accreditation guidelines.

ITIL® is a registered trademark of the PeopleCert group. Used under licence from PeopleCert. All rights reserved.

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: MAZ900 Azure Fundamentals (AZ-900)

If you are new to Azure Administration, consider taking:

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Instructor-led course: MAZ104 Microsoft Azure Administrator (AZ-104)

If you are new to Azure Developer, consider taking:

- Instructor-led course: MAZ204 Developing Solutions for Microsoft Azure (AZ-204)

Please note: In order to access the Azure labs for this course you will need to have a Microsoft Outlook account that has/will not be used to associate with any other corporate Azure subscription. You can set up a new Outlook account here.

What You Will Learn

By the end of this training, you will have gained knowledge and skills in the following areas:

- Design and implement DevOps strategies and CI/CD pipelines.
- Integrate Azure Repos, Pipelines, Artifacts, and GitHub.
- Implement infrastructure as code (ARM, Terraform, Bicep).
- Apply container build strategies with Docker and Kubernetes.
- Manage secrets and configurations using Azure Key Vault.
- Establish monitoring and telemetry with Application Insights.
- Secure DevOps workflows using OWASP and SCA tools.

Training Outline

MODULE 1: Introduction to DevOps

This module explores the key areas that organizations must apply to start their DevOps transformation Journey, change the team's mindset, and define timelines and goals.

By the end of this module, you'll be able to:

- 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

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Introduction
- What is DevOps?
- Explore the DevOps journey
- Identify transformation teams
- Explore shared goals and define timelines
- Knowledge check
- Summary

MODULE 2: Choose the right project

This module helps organizations decide the projects to start applying the DevOps process and tools to minimize initial resistance.

By the end of this module, you'll be able to:

- 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)

- Introduction
- Explore greenfield and brownfield projects
- Decide when to use greenfield and brownfield projects
- Decide when to use systems of record versus systems of engagement
- Identify groups to minimize initial resistance
- Identify project metrics and key performance indicators (KPIs)
- Knowledge check
- Summary

MODULE 3: Describe team structures

This module explores agile development practices and helps to define and to configure teams and tools for collaboration.

By the end of this module, you'll be able to:

- Understand agile practices and principles of agile development
- Create a team and agile organizational structure

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Identify ideal DevOps team members
- Select and configure tools for collaboration

- Introduction
- Explore agile development practices
- Explore principles of agile development
- Define organization structure for agile practices
- Explore ideal DevOps team members
- Enable in-team and cross-team collaboration
- Select tools and processes for agile practices
- Knowledge check
- Summary

MODULE 4: Choose the DevOps tools

This module explores Azure DevOps and GitHub tools and helps organizations define their work management tool and licensing strategy.

By the end of this module, you'll be able to:

- 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

- Introduction
- What is Azure DevOps?
- What is GitHub?
- Explore an authorization and access strategy
- Migrate or integrate existing work management tools
- Migrate or integrate existing test management tools
- Design a license management strategy
- Knowledge check
- Summary

MODULE 5: Plan Agile with GitHub Projects and Azure Boards

This module introduces you to GitHub Projects, GitHub Project Boards and Azure Boards. It explores ways to link Azure Boards and GitHub, configure GitHub Projects and Project views, and manage work with GitHub Projects.

BILGINC IT Academy

info@bilginc.com

bilginc.com

By the end of this module, you'll be able to:

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

- Introduction
- Introduction to GitHub Projects and Project boards
- Introduction to Azure Boards
- Link GitHub to Azure Boards
- Configure GitHub Projects
- Manage work with GitHub Project boards
- Customize Project views
- Collaborate using team discussions
- Agile Plan and Portfolio Management with Azure Boards
- Knowledge check
- Summary

MODULE 6: Introduction to source control

This module introduces you to the basics of source control, exploring benefits and best practices.

By the end of this module, you'll be able to:

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

- Introduction
- Explore DevOps foundational practices
- What is source control?
- Explore benefits of source control
- Explore best practices for source control
- Knowledge check
- Summary

BILGINC IT Academy

info@bilginc.com

bilginc.com

MODULE 7: Describe types of source control systems

This module describes different source control systems such Git and TFVC and helps with the initial steps for Git utilization.

By the end of this module, you'll be able to:

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

- Introduction
- Understand centralized source control
- Understand distributed source control
- Explore Git and Team Foundation Version Control
- Examine and choose Git
- Understand objections to using Git
- Describe working with Git locally
- Knowledge check
- Summary

MODULE 8: Work with Azure Repos and GitHub

This module introduces you to Azure Repos and GitHub and explores ways to migrate from TFVC to Git, and work with GitHub Codespaces for development.

By the end of this module, you'll be able to:

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

- Introduction
- Introduction to Azure Repos
- Introduction to GitHub

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Migrate from TFVC to Git
- Use GIT-TFS
- Develop online with GitHub Codespaces
- Knowledge check
- Summary

MODULE 9: Structure your Git Repo

This module examines Git repositories structure, explains the differences between mono versus multiple repos, and helps you create a changelog.

By the end of this module, you'll be able to:

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

- Introduction
- Explore monorepo versus multiple repos
- Implement a change log
- Knowledge check
- Summary

MODULE 10: Manage Git branches and workflows

This module explores Git branching types, concepts, and models for the continuous delivery process. It helps companies defining their branching strategy and organization.

By the end of this module, you'll be able to:

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

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Introduction
- Explore branch workflow types
- Explore feature branch workflow
- Explore Git branch model for continuous delivery
- Explore GitHub flow
- Explore fork workflow
- Version Control with Git in Azure Repos
- Knowledge check
- Summary

MODULE 11: Collaborate with pull requests in Azure Repos

This module presents pull requests for collaboration and code reviews using Azure DevOps and GitHub mobile for pull request approvals. It helps understanding how pull requests works and how to configure them.

By the end of this module, you'll be able to:

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

- Introduction
- Collaborate with pull requests
- Exercise - Azure Repos collaborating with pull requests
- Examine GitHub mobile for pull request approvals
- Knowledge check
- Summary

MODULE 12: Identify technical debt

This module examines technical debt, complexity, quality metrics, and plans for effective code reviews and code quality validation.

By the end of this module, you'll be able to:

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

- Introduction
- Examine code quality
- Examine complexity and quality metrics
- Introduction to technical debt
- Measure and manage technical debt
- Integrate other code quality tools
- Plan effective code reviews
- Knowledge check
- Summary

MODULE 13: Explore Git hooks

This module describes Git hooks and their usage during the development process, implementation, and behavior.

By the end of this module, you'll be able to:

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

- Introduction
- Introduction to Git hooks
- Implement Git hooks
- Knowledge check
- Summary

MODULE 14: Plan foster inner source

This module explains how to use Git to foster inner sources across the organization, implement Fork

and its workflows.

By the end of this module, you'll be able to:

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

- Introduction
- Explore foster inner source
- Implement the fork workflow
- Describe inner source with forks
- Knowledge check
- Summary

MODULE 15: Manage Git repositories

This module explores how to work with large repositories, purge repository data and manage and automate release notes using GitHub.

By the end of this module, you'll be able to:

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

- Introduction
- Work with large repositories
- Purge repository data
- Manage releases with GitHub Repos
- Automate release notes with GitHub
- Knowledge check
- Summary

BILGINC IT Academy

info@bilginc.com

bilginc.com

MODULE 16: Explore Azure Pipelines

This module introduces Azure Pipelines concepts and explains key terms and components of the tool, helping you decide your pipeline strategy and responsibilities.

By the end of this module, you'll be able to:

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

- Introduction
- Explore the concept of pipelines in DevOps
- Describe Azure Pipelines
- Understand Azure Pipelines key terms
- Knowledge check
- Summary

MODULE 17: Manage Azure Pipeline agents and pools

This module explores the differences between Microsoft-hosted and self-hosted agents, details job types, and configures agent pools. Understand typical situations to use agent pools and how to manage their security.

By the end of this module, you'll be able to:

- 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

- Introduction
- Choose between Microsoft-hosted versus self-hosted agents
- Explore job types

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Introduction to agent pools
- 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 understanding pipeline styles
- Knowledge check
- Summary

MODULE 18: Describe pipelines and concurrency

This module describes parallel jobs and how to estimate their usage. Also, it presents Azure Pipelines for open-source projects, explores Visual Designer and YAML pipelines.

By the end of this module, you'll be able to:

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

- Introduction
- Understand parallel jobs
- Estimate parallel jobs
- Describe Azure Pipelines and open-source projects
- Explore Azure Pipelines and Visual Designer
- Describe Azure Pipelines and YAML
- Knowledge check
- Summary

MODULE 19: Explore continuous integration

This module details continuous integration practice and the pillars for implementing it in the development lifecycle, its benefits, and properties.

By the end of this module, you'll be able to:

BILGINC IT Academy
info@bilginc.com
bilginc.com

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

- Introduction
- Introduction to continuous integration
- Learn the four pillars of continuous integration
- Explore benefits of continuous integration
- Describe build properties
- Enable Continuous Integration with Azure Pipelines
- Knowledge check
- Summary

MODULE 20: Implement a pipeline strategy

This module describes pipeline strategies, configuring them, implementing multi-agent builds, and what source controls Azure Pipelines supports.

By the end of this module, you'll be able to:

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

- Introduction
- Configure agent demands
- Implement multi-agent builds
- Explore source control types supported by Azure Pipelines
- Knowledge check
- Summary

MODULE 21: Integrate with Azure Pipelines

This module details Azure Pipelines anatomy and structure, templates, YAML resources, and how to

use multiple repositories in your pipeline.

By the end of this module, you'll be able to:

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

- Introduction
- Describe the anatomy of a pipeline
- Understand the pipeline structure
- Detail templates
- Explore YAML resources
- Use multiple repositories in your pipeline
- Knowledge check
- Summary

MODULE 22: Introduction to GitHub Actions

In this module, you will learn what GitHub Actions, action flow, and its elements are. Understand what events are, explore jobs and runners, and how to read console output from actions.

By the end of this module, you'll be able to:

- 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

- Introduction
- 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

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Knowledge check
- Summary

MODULE 23: Learn continuous integration with GitHub Actions

This module details continuous integration using GitHub Actions and describes environment variables, artifacts, best practices, and how to secure your pipeline using encrypted variables and secrets.

By the end of this module, you'll be able to:

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

- Introduction
- 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
- Knowledge check
- Summary

MODULE 24: Design a container build strategy

This module helps you plan a container build strategy, explains containers and their structure, introduces Docker, microservices, Azure Container Registry, and related services.

By the end of this module, you'll be able to:

- Design a container strategy

- Work with Docker Containers
- Create an Azure Container Registry
- Explain Docker microservices and containers

- Introduction
- 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
- Knowledge check
- Summary

MODULE 25: Introduction to continuous delivery

This module introduces continuous delivery concepts and their implementation in a traditional IT development cycle.

By the end of this module, you'll be able to:

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

- Introduction
- Explore traditional IT development cycle
- What is continuous delivery?
- Move to continuous delivery
- Understand releases and deployments
- Understand release process versus release
- Knowledge check
- Summary

MODULE 26: Create a release pipeline

BILGINC IT Academy

info@bilginc.com

bilginc.com

This module describes Azure Pipelines capabilities, build and release tasks.

By the end of this module, you'll be able to:

- 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

- Introduction
- Describe Azure DevOps release pipeline capabilities
- Explore release pipelines
- Explore artifact sources
- Choose the appropriate artifact source
- Exercise - select an artifact source
- Examine considerations for deployment to stages
- Exercise - set up stages
- Explore build and release tasks
- Explore custom build and release tasks
- Explore release jobs
- Configure Pipelines as Code with YAML
- Knowledge check
- Summary

MODULE 27: Explore release recommendations

This module explores the critical release strategy recommendations that organizations must consider when designing automated deployments and explains how to define components of a release pipeline and artifact sources, create approves, and configure release gates.

By the end of this module, you'll be able to:

- 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

- Introduction
- Understand the delivery cadence and three types of triggers

- Exercise - select your delivery and deployment cadence
- Explore release approvals
- Exercise - set up manual approvals
- Explore release gates
- Use release gates to protect quality
- Control Deployments using Release Gates
- Knowledge check
- Summary

MODULE 28: Provision and test environments

This module details target environment provisioning, service connections creation process, and test infrastructure setup. Learn how to configure functional test automation and run availability tests.

By the end of this module, you'll be able to:

- 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

- Introduction
- Provision and configure target environments
- Exercise - set up service connections
- 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
- Knowledge check
- Summary

MODULE 29: Manage and modularize tasks and templates

This module describes the creation of task and variable groups and using release variables and stage variables in your pipeline.

By the end of this module, you'll be able to:

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

- Introduction
- Examine task groups
- Exercise - create and manage task groups
- Explore variables in release pipelines
- Understand variable groups
- Exercise - create and manage variable groups
- Knowledge check
- Summary

MODULE 30: Automate inspection of health

This module describes how to automate the inspection of health events, configure notifications in Azure DevOps and GitHub, set up service hooks to monitor pipelines, measure the quality of your release process, and detail release gates for quality purposes. You'll examine release management tools and details about them.

By the end of this module, you'll be able to:

- 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

- Introduction
- Automate inspection of health
- Explore events and notifications
- Explore service hooks
- Exercise - set up service hooks to monitor the pipeline
- 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

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Explore common release management tools
- Knowledge check
- Summary

MODULE 31: Introduction to deployment patterns

This module introduces deployment patterns and explains microservices architecture to help improve the deployment cycle and examine classical and modern deployment patterns.

By the end of this module, you'll be able to:

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

- Introduction
- Explore microservices architecture
- Examine classical deployment patterns
- Understand modern deployment patterns
- Knowledge check
- Summary

MODULE 32: Implement blue-green deployment and feature toggles

This module describes the blue-green deployment process and introduces feature toggle techniques to implement in the development process.

By the end of this module, you'll be able to:

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

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Introduction
- What is blue-green deployment?
- Explore deployment slots
- Exercise - set up a blue-green deployment
- Introduction to feature toggles
- Describe feature toggle maintenance
- Knowledge check
- Summary

MODULE 33: Implement canary releases and dark launching

This module describes deployment strategies around canary releases and dark launching and examines traffic managers.

By the end of this module, you'll be able to:

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

- Introduction
- Explore canary releases
- Examine Traffic Manager
- Understand dark launching
- Knowledge check
- Summary

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

This module introduces A/B test and progressive exposure deployment concepts and explores CI/CD with deployment rings -- ring-based deployment.

By the end of this module, you'll be able to:

- Implement progressive exposure deployment

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

- Introduction
- What is A/B testing?
- Explore CI-CD with deployment rings
- Exercise - Ring-based deployment
- Knowledge check
- Summary

MODULE 35: Integrate with identity management systems

This module describes the integration with GitHub and single sign-on (SSO) for authentication, service principals, and managed service identities.

By the end of this module, you'll be able to:

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

- Introduction
- Integrate GitHub with single sign-on (SSO)
- Explore service principals
- Explore Managed Identity
- Knowledge check
- Summary

MODULE 36: Manage application configuration data

This module explores ways to rethink application configuration data and the separation of concerns method. Explore Azure App Configuration, details Key-value pairs, App Configuration feature management, and integrate Azure Key Vault with Azure Pipelines.

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

By the end of this module, you'll be able to:

- 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

- Introduction
- Rethink application configuration data
- Explore separation of concerns
- Understand external configuration store patterns
- Introduction to Azure App Configuration
- 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
- Knowledge check
- Summary

MODULE 37: Explore infrastructure as code and configuration management

This module describes key concepts of infrastructure as code and environment deployment creation and configuration. Also, understand the imperative, declarative, and idempotent configuration and how it applies to your company.

By the end of this module, you'll be able to:

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

- Introduction

- Explore environment deployment
- Examine environment configuration
- Understand imperative versus declarative configuration
- Understand idempotent configuration
- Knowledge check
- Summary

MODULE 38: Create Azure resources using Azure Resource Manager templates

This module explores Azure Resource Manager templates and their components and details dependencies and modularized templates with secrets.

By the end of this module, you'll be able to:

- 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

- Introduction
- Why use Azure Resource Manager templates?
- Explore template components
- Manage dependencies
- Modularize templates
- Manage secrets in templates
- Deployments using Azure Bicep templates
- Knowledge check
- Summary

MODULE 39: Create Azure resources by using Azure CLI

This module explains Azure CLI to create Azure resources, run templates, and detail Azure CLI commands.

By the end of this module, you'll be able to:

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

- Introduction
- What is Azure CLI?
- Work with Azure CLI
- Exercise - Run templates using Azure CLI
- Knowledge check
- Summary

MODULE 40: Explore Azure Automation with DevOps

This module describes Azure Automation with Azure DevOps, using runbooks, webhooks, and PowerShell workflows. You'll learn how to create and manage automation for your environment.

By the end of this module, you'll be able to:

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

- Introduction
- 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
- Exercise - Create and run a workflow runbook
- Examine checkpoint and parallel processing
- Knowledge check
- Summary

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

MODULE 41: Implement Desired State Configuration (DSC)

This module describes Desired State Configuration (DSC) and its components for implementation. You can exercise how to import, compile and automate your environment creation, and use DSC for Linux automation on Azure.

By the end of this module, you'll be able to:

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

- Introduction
- Understand configuration drift
- Explore Desired State Configuration (DSC)
- Explore Azure Automation State configuration (DSC)
- Examine DSC configuration file
- Exercise - Import and compile
- Exercise - Onboard machines for management
- Explore hybrid management
- Implement DSC and Linux Automation on Azure
- Knowledge check
- Summary

MODULE 42: Implement Bicep

This module explains Bicep and how it integrates with different tools such as Azure CLI and Visual Studio Code for environment deployment configuration.

By the end of this module, you'll be able to:

- 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

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Introduction
- What is Bicep?
- Install Bicep
- Exercise - Create Bicep templates
- Understand Bicep file structure and syntax
- Exercise - Deploy a Bicep file from Azure Pipelines
- Exercise - Deploy a Bicep file from GitHub workflows
- Knowledge check
- Summary

MODULE 43: Introduction to Secure DevOps

This module introduces DevSecOps concepts, SQL injection attacks, threat modeling, and security for continuous integration.

By the end of this module, you'll be able to:

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

- Introduction
- Describe SQL injection attack
- Understand DevSecOps
- Explore Secure DevOps Pipeline
- Explore key validation points
- Explore continuous security validation
- Understand threat modeling
- Exercise threat modeling
- Knowledge check
- Summary

MODULE 44: Implement open-source software

This module explores open-source software and corporate concerns with software components. Also, it explains common open-source licenses, license implications, and ratings.

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

By the end of this module, you'll be able to:

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

- Introduction
- Explore how software is built
- What is open-source software
- Explore corporate concerns with open-source software components
- Introduction to open-source licenses
- Explore common open-source licenses
- Examine license implications and ratings
- Knowledge check
- Summary

MODULE 45: Software Composition Analysis

This module explains Composition Analysis, how to inspect and validate code bases for compliance, integration with security tools, and integration with Azure Pipelines.

By the end of this module, you'll be able to:

- 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

- Introduction
- Inspect and validate code bases for compliance
- Explore software composition analysis (SCA)
- Integrate Mend with Azure Pipelines
- Implement GitHub Dependabot alerts and security updates
- Integrate software composition analysis checks into pipelines
- Examine tools for assess package security and license rate
- Interpret alerts from scanner tools
- Implement security and compliance in an Azure Pipeline
- Knowledge check

- Summary

MODULE 46: Static analyzers

This module introduces the static analyzers SonarCloud and CodeQL in GitHub.

By the end of this module, you'll be able to:

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

- Introduction
- Explore SonarCloud
- Explore CodeQL in GitHub
- Manage technical debt with SonarCloud and Azure DevOps
- Knowledge check
- Summary

MODULE 47: OWASP and Dynamic Analyzers

This module explores OWASP and Dynamic Analyzers for penetration testing, results, and bugs.

By the end of this module, you'll be able to:

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

- Introduction
- Plan Implement OWASP Secure Coding Practices
- Explore OWASP ZAP penetration test
- Explore OWASP ZAP results and bugs

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Knowledge check
- Summary

MODULE 48: Security Monitoring and Governance

This module describes security monitoring and governance with Microsoft Defender for Cloud and its usage scenarios, Azure Policies, Microsoft Defender for Identity, and security practices related to the tools.

By the end of this module, you'll be able to:

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

- Introduction
- 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
- Explore Azure Blueprints
- Understand Microsoft Defender for Identity
- Knowledge check
- Summary

MODULE 49: Explore package dependencies

This module explores dependency management concepts and helps to identify project dependencies. You will learn how to decompose your system, identify dependencies, and package componentization.

By the end of this module, you'll be able to:

BILGINC IT Academy

info@bilginc.com

bilginc.com

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

- Introduction
- 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
- Knowledge check
- Summary

MODULE 50: Understand package management

This module describes package feeds, common public package sources, and how to create and publish packages.

By the end of this module, you'll be able to:

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

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

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

MODULE 51: Migrate consolidating and secure artifacts

This module details package migration, consolidation, and configuration to secure access to package feeds and artifact repositories.

By the end of this module, you'll be able to:

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

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

MODULE 52: Implement a versioning strategy

This module explains versioning strategies for packaging, best practices for versioning, and package promotion.

By the end of this module, you'll be able to:

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

- Introduction
- Understand versioning of artifacts

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Explore semantic versioning
- Examine release views
- Promote packages
- Exercise - Promote a package
- Explore best practices for versioning
- Exercise - Push from the pipeline
- Knowledge check
- Summary

MODULE 53: Introduction to GitHub Packages

This module introduces you to GitHub Packages. It explores ways to control permissions and visibility, publish, install, delete and restore packages using GitHub.

By the end of this module, you'll be able to:

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

- Introduction
- Publish packages
- Install a package
- Delete and restore a package
- Explore package access control and visibility
- Knowledge check
- Summary

MODULE 54: Implement tools to track usage and flow

This module introduces you to continuous feedback practices and tools to track usage and flow, such as Azure Logs Analytics, Kusto Query Language (KQL), and Application Insights.

By the end of this module, you'll be able to:

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

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

- Introduction
- Understand the inner loop
- Introduction to continuous monitoring
- Explore Azure Monitor and Log Analytics
- Examine Kusto Query Language (KQL)
- Explore Application Insights
- Implement Application Insights
- Exercise - Add Application Insights to an ASP.NET core application
- Monitor application performance with Application Insights
- Knowledge check
- Summary

MODULE 55: Develop monitor and status dashboards

This module explains steps to develop monitoring with Azure Dashboards, work with View Designer and Azure Monitor, and create Azure Monitor Workbooks. Also, explore tools to supports monitoring with Power BI.

By the end of this module, you'll be able to:

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

- Introduction
- Explore Azure Dashboards
- Examine view designer in Azure Monitor
- Explore Azure Monitor workbooks
- Explore Power BI
- Build your own custom application
- Knowledge check
- Summary

MODULE 56: Share knowledge within teams

BILGINC IT Academy

info@bilginc.com

bilginc.com

This module describes how to share knowledge within teams, Azure DevOps Wikis, and integration with Azure Boards.

By the end of this module, you'll be able to:

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

- Introduction
- Share acquired knowledge within development teams
- Introduction to Azure DevOps project wikis
- Integrate with Azure Boards
- Share team knowledge using Azure Project Wiki
- Knowledge check
- Summary

MODULE 57: Design processes to automate application analytics

This module helps designing process to Application Insights, explores telemetry and monitoring tools and technologies.

By the end of this module, you'll be able to:

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

- Introduction
- Explore rapid responses and augmented search
- Integrate telemetry
- Examine monitoring tools and technologies
- Knowledge check
- Summary

Design and Implement Microsoft DevOps solutions Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

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

This module examines alerts, blameless retrospectives and creates a just culture. It helps improving application performance, reducing meaningless and nonactionable alerts, and explains server response-time degradation.

By the end of this module, you'll be able to:

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

- Introduction
- Examine when 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
- Knowledge check
- Summary