

Azure DevOps Server

Duration:- 4 Days

Course outline

Module 1: Get started on a DevOps transformation journey

Lessons

- Introduction to DevOps
- Choose the right project
- Describe team structures
- Choose the DevOps tools
- Plan Agile with GitHub Projects and Azure Boards
- Introduction to source control
- Describe types of source control systems
- Work with Azure Repos and GitHub

Lab : Agile planning and portfolio management with Azure Boards

Lab : Version controlling with Git in Azure Repos

After completing this module, students will 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
- 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)
- Understand agile practices and principles of agile development
- Create a team and agile organizational structure

Module 2: Development for enterprise DevOps

Lessons

- Structure your Git Repo
- Manage Git branches and workflows
- Collaborate with pull requests in Azure Repos
- Explore Git hooks
- Plan foster inner source
- Manage Git repositories
- Identify technical debt

Lab : Version controlling with Git in Azure Repos

After completing this module, students will be able to:

- Understand Git repositories
- Implement mono repo or multiple repos
- Explain how to structure Git Repos
- Implement a change log
- Describe Git branching workflows
- Implement feature branches
- Implement GitFlow
- Fork a repo
- Leverage pull requests for collaboration and code reviews
- Give feedback using pull requests

Module 3: Implement CI with Azure Pipelines and GitHub Actions

Lessons

- Explore Azure Pipelines
- Manage Azure Pipeline agents and pools
- Describe pipelines and concurrency
- Explore Continuous integration
- Implement a pipeline strategy
- Integrate with Azure Pipelines
- Introduction to GitHub Actions
- Learn continuous integration with GitHub Actions
- Design a container build strategy

Lab : Configuring agent pools and understanding pipeline styles

Lab : Enabling continuous integration with Azure Pipelines

Lab : Integrating external source control with Azure Pipelines

Lab : Implementing GitHub Actions by using DevOps Starter

Lab : Deploying Docker Containers to Azure App Service web apps

After completing this module, students will be able to:

- Describe Azure Pipelines
- Explain the role of Azure Pipelines and its components
- Decide Pipeline automation responsibility
- Understand Azure Pipeline key terms
- Choose between Microsoft-hosted and self-hosted agents
- Install and configure Azure pipelines Agents
- Configure agent pools
- Make the agents and pools secure
- Use and estimate parallel jobs

Module 4: Design and implement a release strategy

Lessons

- Introduction to continuous delivery
- Create a release pipeline
- Explore release strategy recommendations
- Provision and test environments
- Manage and modularize tasks and templates
- Automate inspection of health

Lab : Creating a release dashboard

Lab : Controlling deployments using Release Gates

After completing this module, students will be able to:

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

Module 5: Implement a secure continuous deployment using Azure Pipelines

Lessons

- 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

Lab : Configuring pipelines as code with YAML

Lab : Setting up and running functional tests

Lab : Integrating Azure Key Vault with Azure DevOps

After completing this module, students will 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
- Differentiate between multi-agent and multi-configuration release job
- 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

- Use and manage task and variable groups

Module 6: Manage infrastructure as code using Azure and DSC

Lessons

- 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 : Azure deployments using Azure Resource Manager templates

After completing this module, students will be able to:

- Understand how to deploy your environment
- Plan your environment configuration
- Choose between imperative versus declarative configuration
- Explain idempotent configuration
- Create Azure resources using ARM templates
- Understand ARM templates and template components
- Manage dependencies and secrets in templates
- Organize and modularize templates
- Create Azure resources using Azure CLI

Module 7: Implement security and validate code bases for compliance

Lessons

- Introduction to Secure DevOps
- Implement open-source software
- Software Composition Analysis
- Static analyzers
- OWASP and Dynamic Analyzers
- Security Monitoring and Governance

Lab : Implement security and compliance in Azure Pipelines

Lab : Managing technical debt with SonarQube and Azure DevOps

After completing this module, students will be able to:

- Identify SQL injection attack
- Understand DevSecOps
- Implement pipeline security
- Understand threat modeling
- Implement open-source software
- Explain corporate concerns for open-source components
- Describe open-source licenses
- Understand the license implications and ratings
- Work with Static and Dynamic Analyzers
- Configure Microsoft Defender for Cloud

Module 8: Design and implement a dependency management strategy

Lessons

- Explore package dependencies
- Understand package management
- Migrate, consolidate, and secure artifacts
- Implement a versioning strategy
- Introduction to GitHub Packages

Lab : Package management with Azure Artifacts

After completing this module, students will be able to:

- Define dependency management strategy
- Identify dependencies
- Describe elements and componentization of a dependency management
- Scan your codebase for dependencies
- Implement package management
- Manage package feed
- Consume and create packages
- Publish packages
- Identify artifact repositories
- Migrate and integrate artifact repositories

Module 9: Implement continuous feedback

Lessons

- 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 Application Insights

Lab : Integration between Azure DevOps and Microsoft Teams

Lab : Sharing Team Knowledge using Azure Project Wikis

After completing this module, students will be able to:

- Implement tools to track feedback
- Plan for continuous monitoring
- Implement Application Insights
- Use Kusto Query Language (KQL)
- Implement routing for mobile applications
- Configure App Center Diagnostics
- Configure alerts
- Create a bug tracker
- Configure Azure Dashboards
- Work with View Designer in Azure Monitor