

**Application Development with Azure DevOps Services**

**Course Number:** AZDO-100
**Duration:** 3 days

**Overview**

Azure DevOps Services is Microsoft’s cloud-based, team-focused services that can help organizations plan smarter, collaborate better, and ship faster with modern developer services. This Application Development with Azure DevOps Services training course teaches attendees how to use Azure DevOps Services to deliver value to their customers faster. Participants learn the platform's various features and how to use them together.

**Prerequisites**

Attendees must have basic programming experience, preferably in C# with Visual Studio Code or Visual Studio 2022.

**Materials**

All Azure DevOps training students receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine

**Objectives**

* Provision Azure DevOps Services for your team or organization
* Onboard developers and stakeholders to Azure DevOps Services
* Track and manage your team's work with Azure Boards
* Use Azure Repos for distributed version control of your team's assets using Git
* Create build pipelines to package up your applications with Azure Pipelines
* Publish your applications and other items to the cloud or on-premises with Azure Pipelines
* Increase quality with Azure Test Plans
* Create, host, and share packages with your team
* Add artifacts to your pipelines with Azure Artifacts

**Outline**

* A Day in the Life of a Developer
	+ Level set of “What is DevOps?”
	+ Overview of Azure DevOps (aka VSTS)
		- Azure Boards
		- Azure Repos & GitHub Repos
		- Azure Pipelines & GitHub Actions
		- Azure Test Plans
		- Azure Artifacts & GitHub Packages
	+ Examine end-to-end workflow
	+ Examine Organization and Team configuration
		- Logins
		- Notifications
		- The core features that every team member needs to know
* Azure Boards: Plan, Track, and Discuss Work Across Your Teams
	+ What does every team member need to know?
	+ Organizing and refining the Product Backlog
	+ Tools in Azure Boards
		- The Product Backlog
		- Kanban boards
		- Task Boards
	+ Dependencies, types, and related risks
	+ Planning and executing a Sprint
		- Limiting work in progress (WIP)
		- Working in small batches
		- Creating and accepting a definition of “Done”
	+ Using queries, charts, and dashboards for basic reporting
	+ “Just enough project management” to support full DevOps traceability
* Azure Repos & GitHub Repos: Using Git Effectively
	+ Centralized vs. decentralized version control
	+ Defining and managing repos
	+ Don’t fear the command-line
	+ Working with branches
	+ Pull Requests
	+ Using Code Search
* Azure Pipelines & GitHub Actions: Building Quality In
	+ Defining Quality Gates
	+ Azure Pipelines for Builds
	+ Understanding and Managing Agents
	+ YAML Build Definitions
	+ Testing with Build
		- Unit Testing
		- Code Coverage
		- Test Impact Analysis
	+ Managing and Sharing Build Definition
* Azure Pipelines: Releasing to the World
	+ Understanding deployment models
	+ Azure Pipelines for Releases
	+ YAML Release Definitions
	+ Service Connections
	+ Stages and environments
	+ Defining approval processes and quality gates
	+ Deployment Groups and Targets
	+ Managing and Sharing Release Definitions
* Infrastructure & Configuration Management
	+ Infrastructure as Code
	+ Create Azure Resources with ARM Templates
	+ Create Azure Resources with Bicep Templates
	+ Create Azure Resources with Azure CLI & PowerShell
	+ Implement Desired State Configuration
	+ Azure Automation with DevOps
* Azure Artifacts & GitHub Packages: Sharing Code Effectively
	+ Why and what of the service
	+ Integrating with Azure Pipelines
	+ Designing your repos to better support sharing
	+ Updating your workflow to support collective ownership mindset
	+ Branching strategies
	+ Applying Open-Source Software principles to internal development
* Automated Testing
	+ Building Integration Tests
	+ Automated UI Testing for Web Apps
	+ Integrating Automated Tests with Builds and Releases
* Feedback: User Acceptance Testing, Monitoring, and Analytics
	+ Continuous Feedback
	+ Involving Stakeholders
	+ Running UAT Tests
	+ Application Insights
	+ Quality Tracking and Reporting
	+ Azure DevOps Analytics
* Conclusion