

Azure DevOps

Course ISI-1504C 4 Days Instructor-led, Hands on

Course Information

This 4-day, instructor-led hands-on course teaches Azure DevOps, formerly known as Team Foundation (TFS and VSTS). Azure DevOps is a set of collaborative software development tools, hosted on-premises and in the cloud. Azure DevOps integrates with your existing IDE or editor, enabling your cross-functional team(s) to work effectively on projects of all sizes. Azure DevOps works for any language, and on any platform. Azure DevOps has everything you need to turn an idea into a working piece of software. You can plan your project with Agile tools, you can manage your test plans, version your code using Git, and deploy your solution using an incredible cross-platform CI-CD system, all while getting full traceability and visibility across your development activities.

Target Audience

This course is aimed for software professionals including developers, testers, architects, configuration analysts and release managers who want to understand the capabilities of Azure DevOps.

Prerequisites

Users must have knowledge in using source control and Visual Studio.

Course Outline

Module 1: DevOps Principles and Azure DevOps Project Management

Azure DevOps Overview

- Introducing DevOps
- Understanding DevOps principles
 - Principle 1 – Customer-centric action
 - Principle 2 – Create with the end in mind
 - Principle 3 – End-to-end responsibility
 - Principle 4 – Cross-functional autonomous teams
 - Principle 5 – Continuous improvement
 - Principle 6 – Automate everything
- Introducing Azure DevOps key concepts
 - Plan
 - Develop
 - Deliver
 - Operate

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Continuous integration and continuous delivery (CI/CD)
 - Agile development support
 - Version control
 - Infrastructure as Code
 - Configuration Management
 - Monitoring
- Discovering Azure DevOps services
 - Azure Boards
 - Azure Repos
 - Azure Pipelines
 - Azure Test Plans
 - Azure Artifacts
 - Extension Marketplace
- Introducing the scenarios
 - Creating the starter project

Managing Projects with Azure DevOps Boards

- Technical requirements
- Understanding processes and process templates
- Creating an organization
- Creating a project
- Creating and managing project activities
 - Work Items
 - Backlogs
 - Boards
 - Sprints
 - Queries

Module 2: Source Code and Builds

Source Control Management with Azure DevOps

- Technical requirements
- Understanding SCM
- Exploring branching strategies
 - GitHub Flow
 - GitLab Flow
 - Git Flow
- Handling source control with Azure DevOps
 - Cloning a remote repository
 - Importing a GitHub repository into Azure DevOps
 - Working with commits, pushes, and branches
 - Protecting branches with policies
 - Cross-repo policies
- Working with pull requests
- Creating a pull request from the Azure DevOps pull request page

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Creating a pull request from a work item
 - Creating a pull request after pushing a branch
 - Creating a pull request from Visual Studio Code or Visual Studio
- Handling a pull request
- Tagging a release

Understanding Azure DevOps Pipelines

- Technical requirements
- Implementing a CI/CD process
- Overview of Azure Pipelines
- Understanding build agents
 - Microsoft-hosted agents
 - Self-hosted agents
 - When to use a Microsoft-hosted or a self-hosted agent
- Overview of the YAML language
 - Scalars
 - Collections and lists
 - Dictionaries
 - Document structure
 - Complex object definition
- Creating a build pipeline with Azure DevOps
 - Pipeline definition with the classic editor
 - YAML pipeline definition
- Retention of builds
- Multi-stage pipeline
- Building a pipeline with GitHub repositories
 - Executing jobs in parallel in an Azure Pipeline
 - Agents on Azure Container Instances
- Using container jobs in Azure Pipelines

Running Quality Tests in a Build Pipeline

- Technical requirements
- Benefits of automatic testing
- Introduction to unit testing
- Running unit tests in a build pipeline
 - Downloading the source code
 - Creating the pipeline
- Introduction to code coverage testing
- Performing code coverage testing
- Assigning test results to work items
- Introduction to Feature Flags
- Using Feature Flags to test in production
 - Creating a new .NET Core application

Hosting Your Own Azure Pipeline Agent

- Technical requirements
- Azure pipeline agent overview
- Understanding the types of agents in Azure Pipelines
 - Microsoft-hosted agents
 - Self-hosted agents
- Planning and setting up your self-hosted Azure pipeline agent
 - Choosing the right OS/image for the agent VM
 - OS support and pre-requisites for installing an Azure Pipelines agent
 - Creating a VM in Azure for your project
 - Setting up the build agent
- Updating your Azure pipeline to use self-hosted agents
 - Preparing your self-hosted agent to build the Parts Unlimited project
 - Running the Azure pipeline
- Using containers as self-hosted agents
 - Setting up Windows containers as Azure pipeline agents
 - Setting up Linux containers as Azure Pipelines agents
 - Using Azure Container Instances as agents
 - Environment variables
 - Planning for scale
 - Creating an Azure VM scale set
 - Setting up Azure pipeline agents with VM scale set

Module 3: Artifacts and Deployments

Using Artifacts with Azure DevOps

- Technical requirements
- Introducing Azure Artifacts
- Creating an artifact feed with Azure Artifacts
- Producing the package using a build pipeline
 - Adding the sample project to the PartsUnlimited repository
 - Creating the build pipeline
- Publishing the package to the feed from a build pipeline
 - Setting the required permissions on the feed
- Consuming the package in Visual Studio from the Artifacts feed
- Scanning for package vulnerabilities using WhiteSource Bolt

Deploying Applications with Azure DevOps

- Technical requirements
- An overview of release pipelines
- Creating a release pipeline with Azure DevOps
 - Creating the Azure DevOps release
 - Configuring the release pipeline triggers for continuous deployment
- Creating a multi-stage release pipeline

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Using approvals and gates for managing deployments
 - Creating approvals
 - Using gates to check conditions
 - Using deployment groups
- YAML release pipelines with Azure DevOps

Module 4: Advanced Features of Azure DevOps

Integrating Azure DevOps with GitHub

- Technical requirements
- An overview of Azure DevOps and GitHub integration
- Integrating Azure Pipelines with GitHub
 - Setting up Azure Pipelines and GitHub integration
 - Testing continuous integration
 - Adding a build Status badge
- Integrating Azure Boards with GitHub
 - Setting up Azure Boards and GitHub integration
 - Adding an Azure Boards Status badge
 - Linking Azure Boards work items to GitHub objects
 - Updating work items from GitHub
- Overview of GitHub Actions

Using Test Plans with Azure DevOps

- Technical requirements
- Introduction to Azure Test Plans
- Exploratory testing
 - Installing and using the Test & Feedback extension
- Planned manual testing
- Test plans, test suites, and test cases
 - Managing test plans, test suites, and test cases
- Running and analyzing a manual test plan

Real-World CI/CD Scenarios with Azure DevOps

- Technical requirements
- Setting up a CI/CD pipeline for .NET-based applications
 - Introduction to the sample application
 - Preparing the pre-requisite Azure infrastructure
 - Setting up an Azure DevOps project
- Setting up a CI/CD pipeline for a container-based application
 - Introduction to the sample app
 - Setting up the required infrastructure
 - Setting up Azure Repos for the voting application
 - Setting up the CI pipeline
 - Setting up the CD pipeline



- Simulating an end-to-end CI/CD experience
- Azure Architecture Center for DevOps