

**Microservices Development in Windows**

**Course Number:** MSV-102WA  
**Duration:** 2 days

**Overview**

This Microservices Development in Windows training course teaches attendees how to migrate applications to microservice architected solutions with containerization for Microsoft development and deployment environments. Participants also learn architecture for .NET, CI/CD with Visual Studio Team Services, Visual Studio Tooling, Azure Container Services, Kubernetes, Docker, and more.

**Prerequisites**

No prior experience is presumed.

**Materials**

All Microservices 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 work well.

**Objectives**

* Confidently use the stack outlined in the course
* Understand the various key components
* Apply the knowledge to migrate applications to microservice architected solutions with containerization for Microsoft development and deployment environments

**Outline**

* Introduction
* Microservice Development
  + The Microservices Architecture Design Principles
  + Decentralized Processing
  + Business Domain-Centric Design
  + Crossing Process Boundary is Expensive!
  + Designing for Failure
  + Fault Injection During System Testing
  + Distributed Transactions
  + Managing Distributed Services with Docker & Kubernetes (OpenShift)
  + Microservices and their relationship to the front-end
  + Rich Client Applications
  + Single Page Applications (SPA)
  + The Building Blocks of a Fault-tolerant Application
  + Example of Microservices in Their Purest Form: AWS Lambdas
  + Example of Traditional Enterprise Application Architecture
  + Example of Microservices Architecture
* REST Services
  + Many Flavors of Services
  + Understanding REST
  + Principles of RESTful Services
  + Create
  + Retrieve
  + Update
  + Delete
  + Client Generated ID
  + SOAP Equivalent Examples
  + JSON
  + REST vs SOAP Communication
  + Restful API Design
  + Versioning
  + Documentation
  + Security
  + Additional Resources
* Getting started with asp.net Core
  + Getting Started with ASP.NET Core
  + Configuration and Middleware Pipeline
  + Controllers and Views
  + RESTful Services with Web API
  + Data Access with Entity Framework Core
* Serverless architecture with Azure
  + What is a Serverless Architecture?
  + Azure Functions
  + WebJobs SDK
  + WebJobs Core
  + WebJobs Extensions
  + Azure App Service
  + Azure Resource Manager
  + Understanding Azure Functions usage
  + Azure Container Service
  + Azure Service Fabric
  + Microsoft Application Insights
  + Microsoft Operations Management Suite
* Azure container service w/ Kubernetes
  + What is Kubernetes?
  + What Is a Container?
  + Microservices and Orchestration
  + Microservices and Infrastructure-as-Code
  + Kubernetes Container Networking
* Kubernetes: from the firehose
  + Masters
  + Nodes
  + Pods
  + Namespaces
  + Resource Quota
  + Authentication and Authorization
  + Routing
  + Registry
  + Storage Volumes
  + Microservices, Linking, and Catalogs
* Docker: Bring on the Whale
  + What is Docker
  + Docker ecosystem
  + Docker concepts
  + Docker Architecture
  + Microservice encapsulation
  + Secure microservices using Docker
  + Agility, Portability, and Control
  + .NET Core or .NET Framework for Docker
* Microsoft Docker Tooling
  + Using Visual Studio Tools for Docker
  + Configuring your local environment
  + Using Docker Tools in Visual Studio 2015
  + Using Docker Tools in Visual Studio 2017
  + Using Windows PowerShell commands in a DockerFile to set up Windows Containers
  + DevOps workflow for a Docker application
  + Inner-loop development workflow
  + Source-Code Control integration and management with Visual Studio Team Services and Git
  + Build, Continuous Integration, and Test with Visual Studio Team Services and Docker
  + Continuous Delivery, Deploy
  + Run and manage
  + Monitor and diagnose
* Operational Readiness
  + Monitoring, Logging, and Resource Management
  + Monitoring Compute Resources
  + Checking Pod Health
  + Logging
  + Secrets & Vaulting
* Application Modernization
  + What is Application Modernization?
  + Typical App Modernization Projects
  + Why Modernize?
  + Goals for Modernization
  + Twelve-factor Application Microservices
  + Maintaining State
  + Cloud Service Fabric
* Security in microservices
  + Role-based
  + Claim-based
  + Session Cookie
  + HTTP Basic
  + JWT
* Conclusion