

Developing Applications : Microservices with Containers

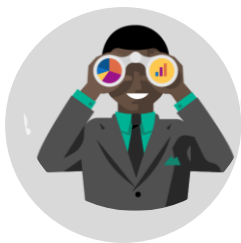
WorkshopPLUS

Deep-Dive into Microservices and Containers

Do you have expensive, recurring issues with your large monolithic applications, such as poor maintainability, long time-to-market, unmanageable legacy code and spaghetti architecture?

Step into the new world of microservices, an increasingly popular architectural style, in which a system is developed as a suite of small, composable services, each built around a specific business capability and independently deployable. Embracing cross-platform, each can be written in a different programming language, use different data storage technologies and scale as required.

OUTCOMES



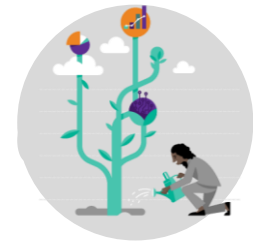
Way Forward

Take your team to the next level with microservice-based systems, leveraging container technologies



Skills

Build architectural and development expertise through classroom education and hands-on labs



Best Practices

Identify, understand and apply widely-accepted patterns, principles and best practices

CAPABILITIES

01 GET READY

Deepen your understanding of a microservice-based architecture

02 LEARN BY EXAMPLE

Assemble an end-to-end microservices reference application with closely-guided scenarios

03 BUILD APPS

Validate your decisions to construct microservices and ensure they leverage best patterns and practices

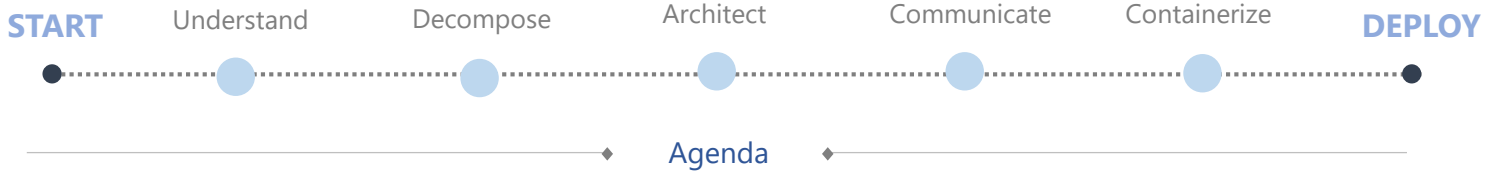
Audience

Developers Architects Technical Leaders

A four-day deep-dive into the design and implementation of microservice-based systems in the cloud. Content can be reduced, expanded or customized to meet your specific development needs.

AGENDA

Duration: 3 days



Module 1: Introducing Microservices

We define microservices: What they are? How they work? Their benefits and challenges? The problems they solve. Importantly, we identify systems that would warrant this architectural approach.

Module 2: Modeling Microservices

We examine best practices for identifying and scoping microservices. What are the drivers for partitioning them? How granular should they be? How do I approach replatforming a legacy monolithic system to a microservice architecture?

Module 3: Cloud-Native Infrastructure

Discussion of resource groups, Azure SQL Database, Azure Storage, Azure Service Bus, Cosmos DB, Configuration and Secrets and how all these services are used with microservices.

Module 4: Architecting Microservices

We explore a microservice architecture, its characteristics and design. Emphasis is given to widely-accepted patterns and principles that benefit this type of architecture.

Module 5: Microservice Communication

We explore how clients communicate with microservices and how microservices collaborate with one another. Request/Response, Publish/Subscribe, API Gateways and more. Emphasis is on the trade-offs among messaging patterns..

Module 6: Distributed Data

We consider best practices for managing distributed data. Emphasis is given to managing consistency and the trade-offs around implementing cross-service queries. We cover CQRS and Event Sourcing.

Module 7: DDD and CQRS

Learn how Domain Driven Design (DDD) and Command and Query Response Segregation (CQRS) are used with microservices.

Module 8: Resiliency and Idempotency

Gain and understanding of resiliency using the Polly framework. Discuss handling of failure using Retry and Circuit Breaker design. Also discuss the importance of idempotent operations for microservices

Module 9: Containers for Developers

We explore software containers from a developer's perspective. Emphasis is on images, container creation and Docker tooling. Special attention is given to defining a multi-container system with Docker Compose.

Module 10: Orchestration

Orchestrators automate the provisioning and management of containerized microservice workloads. We explore the importance of orchestrators as an enabler for working with microservices.

Module 11: Deploying Microservices to Kubernetes

We deploy a set of containerized microservices to Azure Kubernetes Service. Emphasis is given to the orchestration and management features of Kubernetes along with decision of criteria of when to use it