

Workshop

Microservices with .NET Core

– Building, Managing & Orchestrating

Duration: 3-5 days

Attendees: Up to 16 (2 trainers)

Course Description

After this workshop, you will get a **vast understanding of building distributed applications** using .NET Core. By following the good patterns and practices, covering the various aspects of event-driven approach, discussing in-depth topics such as clean architecture, asynchronous integration, internal communication, monitoring, distributed tracing, testing, deployment, containerization and orchestration, **you shall find the practical solutions** to the most common challenges when it comes to building the microservices.

Moreover, you will get to know how to make use of some of the most acknowledged open source tools being part of Cloud Native Computing Foundation in order to make your solution **cloud-agnostic**, meaning that it can run on-premise (bare metal) as well as in chosen cloud provider, **avoiding any potential vendor lock-in**.

The workshops are 30/70 between theory and practice. Many of the mentioned aspects are exercised practically, as it's important to gain the required knowledge and see the actual code in action.

Notice: Workshop might last between 3-5 days - in such case, we have time to cover all of the topics and go through all the exercises. In the case of 3- or 4-days workshops, some topics are omitted for brevity, depending on the client's needs.

What the attendee will learn:

- Understand how to design the microservices and define their boundaries
- Implementing distributed applications using .NET Core, clean architecture and CQRS
- Approaching the event-driven architecture, internal communication, service discovery, load balancing, logging, monitoring, distributed tracing and many more
- How to build a cloud-agnostic solution by integrating with CNCF open-source tools
- Dealing with distributed transactions, Saga, unit/integration/contract testing
- Dockerize the services, and orchestrate them with Kubernetes and service mesh

Program Outline:

Day 1

Fundamental concepts

- Pros & cons of distributed applications
- Designing microservices and finding their boundaries

Hands-on implementation

- An overview of the clean architecture
- Building lightweight applications using .NET Core
- A practical approach to CQRS

Asynchronous integration

- Event-driven architecture in-depth
- Handling the asynchronous events with RabbitMQ message broker
- CAP and eventual consistency implications

Day 2

Internal communication

- API Gateway and its role
- Immediate consistency implications
- Challenges of HTTP communication
- Service discovery with Consul and load balancing with Fabio

Asynchronous API

- Processing asynchronous commands
- Tracking the state of ongoing requests
- Using WebSockets and gRPC to handle real-time notifications

Microservices with .NET Core – Building, Managing & Orchestrating

Day 3

Distributed transactions

- Dealing with complex business processes using 2PC, event choreography, and Saga
- Saga and Process manager implementation

Testing the microservices

- Unit testing (domain, handlers, etc.)
- Integration testing (in-memory)
- Contract testing with Pact library

Day 4

Tooling

- Secured configuration with Vault and logging with Seq
- Monitoring with Prometheus and Grafana
- Distributed tracing with Jaeger

Docker fundamentals

- An introduction to the containers
- Building images with custom Dockerfiles

Day 5

CI & CD

- Advanced Docker topics (networking, registry, etc.) and Docker-compose
- Setting up an automated build environment

Orchestration

- Kubernetes basics – pods, services, deployments, ingress
- Setting up a Kubernetes cluster with Rancher
- Handling cross-cutting infrastructural concerns using Istio service mesh

Microservices with .NET Core – Building, Managing & Orchestrating

Target Audience

Any .NET/.NET Core developer should benefit from this workshop! Starting from the basics of distributed applications, we go through every aspect of designing, implementing, testing, managing and deploying the microservices. Not only the novice programmers, but the advanced software engineers will learn a lot, as there are many challenges that have to be tackled when working with microservices.

Audience Requirements

Familiarity with .NET Core ecosystem, especially at least average knowledge of C# and Web API.

About Trainers

- **Piotr Gankiewicz** - Microsoft MVP, Bottega IT Minds trainer, a software engineer at Nethermind working on .NET Core Ethereum Blockchain client implementation. Blogger, speaker, open-source contributor and co-founder of DevMentors.io platform with free programming courses.
- **Dariusz Pawlukiewicz** - Microsoft MVP, Bottega IT Minds trainer, a software engineer at Connectis_ working with microservices for EG company. Blogger, speaker, open-source contributor and co-founder of DevMentors.io platform with free programming courses.