BO·TT·EGA
IT minds

Training program:

Microservices .NET Core - Part II: Deployment architecture

Info:

Name:	**Microservices .NET Core - Part II: Deployment architecture**
Code:	**NET-arch-ms2**
Category:	.NET Architecture
Target audience:	developers architects devops
Duration:	3 days
Format:	30% lecture / 70% ćwiczenia

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.

BO·TT·EGA
IT minds

Training program

1. Internal services communication

1.1. Integration vs communication

1.2. Service registry and discovery

1.3. Load balancing

2. API gateway

2.1. API Gateway Responsibilities

2.2. Synchronous vs asynchronous gate

3. Distributed business transactions

3.1. Overview of available patterns

3.2. Saga + Process manager in practice

4. Authentication in a distributed system

4.1. Stateless Authentication

4.2. Authorizing access to resources

5. Testing microservices

5.1. Unit tests

5.2. End-to-end tests

5.3. Integration tests

5.4. Contract tests

5.5. Performance tests

6. System observability

6.1. Centralized logging

6.2. Monitoring and metrics

6.3. Distributed tracing

7. Security

7.1. Dynamic generation of credentials

7.2. PKI certificate rotation

7.3. Digital signature

8. DevOps

8.1. Docker - containerization

8.2. Kubernetes fundamentals

8.3. Sidecar pattern

8.4. Deployment and orchestration

8.5. Service mesh