Curso ATDEV340 - Desenvolvimento de Aplicações Distribuídas com Containers

Duração: 21h

Introdução

As aplicações modernas são cada vez mais distribuídas, por exigência de escalabilidade, pelas plataformas cloud e pela crescente complexidade dos sistemas. Com essa complexidade vêm várias perguntas: porquê usar containers? Como os construo e depuro? Como comunicam os serviços entre si? Como torno a aplicação mais resiliente? Que opções tenho para a colocar em produção? Esta formação responde a estas perguntas e acompanha os participantes no caminho para construir aplicações distribuídas fiáveis e escaláveis em projetos .NET reais.

Esta formação incide sobre o desenvolvimento de aplicações distribuídas, não sobre o deployment. Para containerização e deployment, recomenda-se o curso MAAZCK «Azure | Gerir Containers e Kubernetes».

Público-alvo

Este curso destina-se a programadores .NET experientes e a arquitetos de software que pretendam modernizar as suas competências.
É ideal para quem queira dominar as complexidades da containerização e dos padrões cloud-native sem se perder na parafernália de infraestrutura.

No final do curso

Após a conclusão deste curso, os formandos deverão ser capazes de:
• Compreender os desafios das aplicações distribuídas.
• Colocar aplicações ASP.NET Core em containers com Docker.
• Construir aplicações distribuídas com Aspire e integrações plugáveis.
• Implementar comunicação síncrona e assíncrona entre serviços.
• Aplicar padrões de orquestração e publicar aplicações em containers.

Pré-requisitos

Experiência de desenvolvimento em .NET.

Exames de certificação

(não existem exames)

Conteúdo programático em detalhe

Introdução às aplicações distribuídas
• A arquitetura de aplicações ao longo dos anos
• Containers: a porta de entrada para as aplicações distribuídas
• Pontos fortes e desafios da arquitetura de aplicações moderna
• O Aspire vem em auxílio
Containers
• Porquê containers?

- Containers vs. máquinas virtuais
- Containers Linux e Windows
- Opções de orquestração de containers

Docker

- Executar o Docker
- Containers, imagens e registries
- Construir uma imagem
- Mapeamento de portas e volumes
- Orquestração de containers com Docker Compose
- LAB: Node.js simples em Docker

Colocar aplicações ASP.NET Core em containers

- Imagens Docker oficiais do .NET
- Colocar dependências em containers
- Ferramentas
- LAB: carregador de ficheiros

Construir aplicações distribuídas com Aspire

- Orquestração em C# com o AppHost
- Aspire Dashboard: tenha a melhor experiência de desenvolvimento possível, de borla
- Aspire CLI
- Ligar os seus projetos entre si com service discovery
- Deve o ambiente de desenvolvimento tentar mesmo imitar o de produção?
- LAB: construir uma aplicação distribuída com Aspire

Enriquecer a aplicação com integrações plugáveis

- O que são as integrações do Aspire?
- Integrações mais usadas: caching, mensagens, armazenamento e outras
- Alojar e consumir integrações
- Configurar integrações
- LAB: acrescentar integrações existentes à sua aplicação

Comunicação síncrona

- Problemas da comunicação direta
- Rate limiting
- Timeouts totais e por tentativa
- Retry
- Circuit breaker
- Sidecar pattern
- LAB: personalizar os handlers de resiliência do HttpClient

Comunicação assíncrona

- Padrões de comunicação
- Comunicação baseada em eventos
- Consistência eventual
- LAB: consistência eventual com RabbitMQ

Padrões de orquestração de containers

- Sidecar pattern: estender comportamento e separar responsabilidades
- Work queue pattern: produtor-consumidor e processamento em batch

- Init containers: tratar tarefas de arranque e dependências
- Desacoplar sistemas para ganhar reutilização e portabilidade
- LAB: trabalhar com init containers e o work queue pattern

Deployment de aplicações em containers

- Gestão de imagens em container registries
- Opções de deployment: ACA, ACI e Kubernetes
- Tirar partido do manifesto do Aspire
- Integração e entrega contínuas (CI/CD)
- LAB: publicar no Azure Container Instances