

Architektura mikroserwisów w Pythonie



Kategoria	Czas trwania	Termin	Cena
Software Design	24h / 3 dni	ustalamy indywidualnie	ustalamy indywidualnie

Program szkolenia:

Poniżej przedstawiamy przykładowy program szkolenia, który może zostać zmodyfikowany zgodnie z oczekiwaniami oraz poziomem grupy szkoleniowej. Przed przygotowaniem docelowego programu szkolenia, przeprowadzamy rozmowę techniczną, w której bierze udział trener oraz osoba techniczna lub cały zespół developerów reprezentujący klienta, w celu ustalenia szczegółów szkolenia.

Dzień 1: Wprowadzenie i Podstawy Mikroserwisów

Architektura monolityczna i wprowadzenie do mikroserwisów

- Cechy architektury monolitycznej – korzyści i ograniczenia.
- Główne założenia mikroserwisów oraz omówienie różnic między monolitem a mikroserwisami.

Przegląd popularnych architektur mikroserwisowych

- Omówienie kluczowych architektur, takich jak:
 - ❖ API Gateway
 - ❖ CQRS
 - ❖ Event Sourcing
 - ❖ DDD
 - ❖ Wzorzec Sidecar
- Dyskusja na temat różnych scenariuszy wdrażania i wyzwań każdej z architektur.

Wyznaczanie granic odpowiedzialności

- Analiza domeny, wyznaczanie granic odpowiedzialności, identyfikacja komponentów.
- Warsztat: Projektowanie podziału monolitu na mikroserwisy
- ❖ Praca nad przykładem gotowej, złożonej aplikacji monolitycznej – identyfikacja głównych funkcji i przepływów danych.
- ❖ Wyznaczanie granic odpowiedzialności w oparciu o przykłady biznesowe.
- ❖ Definiowanie sposobu komunikacji między wyznaczonymi mikroserwisami.

Dzień 1 c.d.

Aplikacje mikroserwisowe w języku Python

- Omówienie wzorców projektowych w Pythonie dla mikroserwisów.
- Przegląd frameworków – wnioski z pracy z językiem Python
- Gotowe szablony aplikacji
- ❖ API
- ❖ workery (procesy wielowątkowe)
- ❑ obsługa wątków pod optymalizację procesu wdrożeń
- ❑ zarządzanie bibliotekami
- ❑ obsługa zdarzeń i logów
- Konfiguracja środowiska Python i przygotowanie do warsztatów.
- ❖ zarządzanie bibliotekami
- ❖ środowisko wirtualne i jego znaczenie w systemach skonteneryzowanych
- ❖ migracje bazodanowe

Dzień 2: Proces Migracji i Podstawy Orkiestracji

Proces migracji z monolitu do mikroserwisów

- Omówienie etapów migracji:
- ❖ Automatyzacja testów
- ❖ Określenie wszystkich planowanych i nieplanowanych środowisk wdrożeniowych
- ❖ Wyznaczenie granic i podział monolitu
- ❖ Zaplanowanie kształtu repozytoriów i ich optymalizacja pod kątem CI/CD
- ❖ Określenie możliwych procesów wdrożeniowych (środowiska DMZ vs prywatne)
- ❑ wdrożenia przez serwery przesiadkowe
- ❑ pełna automatyzacja z wykorzystaniem GitOps
- Warsztat: Weryfikacja migracji poprzez implementację podstawowych usług w Pythonie
- ❖ Przekształcenie przykładowych funkcji monolitu w osobne mikroserwisy.
- ❖ Uruchomienie usług lokalnie oraz walidacja podziału odpowiedzialności.

Dzień 2 c.d.

Konteneryzacja aplikacji monolitycznej i mikroserwisowej

- Podstawy Docker i konteneryzacji, zasady tworzenia lekkich i bezpiecznych obrazów.
- ❖ Przegląd typowych błędów i pułapek
- ❖ Dobre praktyki
- Warsztat: Konteneryzacja aplikacji w Pythonie
- ❖ Tworzenie Dockerfile dla różnych komponentów mikroserwisowych.
- ❖ Testowanie kontenerów lokalnie oraz omówienie wpływu błędów na działanie systemu

Orkiestracja przy użyciu Kubernetes

- Wprowadzenie do Kubernetes, podstawowe komponenty (Pod, Deployment, Service).
- Managed Kubernetes Service vs Rancher vs K3s
- Warsztat: Tworzenie i zarządzanie mikroserwisami w Kubernetes
- ❖ Zadanie: Konfiguracja prostego środowiska na Kubernetes, wdrażanie i monitorowanie mikroserwisów.
- ❖ Praca z konfiguracjami YAML dla Deployment i Service.
- Omówienie metodyki GitOps

Dzień 3: Monitorowanie, Logi, CI/CD oraz „12 Factor App”

Monitoring, tracing, przetwarzanie logów i alerting

- Narzędzia do monitoringu i logowania w środowiskach mikroserwisowych (Prometheus, Grafana, Jaeger).
- Warsztat: Konfiguracja monitoringu i alertowania dla mikroserwisów na Kubernetes
- ❖ Integracja z Prometheus i Grafana w celu monitorowania wydajności mikroserwisów.
- ❑ Docker, z wykorzystaniem gotowych skryptów Ansible
- ❑ Kubernetes, z wykorzystaniem gotowych skryptów Terraform
- ❖ Konfiguracja distributed tracing oraz logowanie z wykorzystaniem narzędzi typu open-source.

Dzień 3 c.d.

Proces deploymentu w oparciu o CI/CD

- Zasady CI/CD dla mikroserwisów, podstawowe konfiguracje (Jenkins).
- Warsztat: Konfiguracja prostego CI/CD
- ❖ Tworzenie przykładowego pipeline dla mikroserwisów – automatyzacja budowania, testowania i wdrażania.
- ❖ Wersjonowanie aplikacji w oparciu o semver – zarządzanie wersjonowaniem i wdrożenie kilku wersji usługi.

Zasady „12 Factor App” w mikroserwisach

- Omówienie 12 zasad aplikacji zorientowanych na mikroserwisy.
- Warsztat: Ocena istniejących mikroserwisów względem „12 Factor App”
- ❖ Przegląd i analiza istniejących serwisów pod kątem zgodności z zasadami.
- ❖ Dostosowanie wybranych usług do standardu „12 Factor App” w praktyce.

KONTAKT


Jesteś zainteresowany dedykowanym
szkoleniem dla Twojej firmy?

Skontaktuj się z Przemkiem!



PRZEMYSŁAW WOŁOSZ

Key Account Manager

 (+48) 730 830 801

 przemyslaw.wolosz@infoShareAcademy.com