

# Korišćenje više servisa istovremeno kao deo istog projekta

Zlatko Hajdarevic

17. jul 2024.

# Sadržaj

Uvod u mikrousluge

Korišćenje više servisa

Integracija više servisa

Primeri upotrebe više servisa

Izazovi u korišćenju više servisa

Alati i tehnologije za rad sa više servisa

Zaključak

# Uvod u mikrousluge

- ▶ Mikrouslužna arhitektura (Microservices Architecture) je pristup razvoju softvera gde se aplikacija sastoji od malih, nezavisnih servisa.
- ▶ Svaki servis obavlja specifičan zadatak i komunicira sa drugim servisima putem dobro definisanih API-ja.
- ▶ Prednosti mikrouslužne arhitekture:
  - ▶ Skalabilnost.
  - ▶ Fleksibilnost u razvoju i implementaciji.
  - ▶ Lakše održavanje i testiranje.

# Korišćenje više servisa

- ▶ U sklopu jednog projekta, može biti potrebno koristiti više različitih servisa.
- ▶ Svaki servis može biti odgovoran za određeni deo funkcionalnosti:
  - ▶ Autentifikacija i autorizacija.
  - ▶ Upravljanje korisnicima.
  - ▶ Upravljanje proizvodima.
  - ▶ Obrada plaćanja.
- ▶ Servisi komuniciraju međusobno putem API poziva ili poruka.

# Integracija više servisa

- ▶ Postoji nekoliko metoda za integraciju više servisa u jedan projekat:
  - ▶ **REST API:** Servisi koriste HTTP protokol za komunikaciju putem REST API-ja.
  - ▶ **gRPC:** Servisi koriste HTTP/2 protokol i Protobuf za brzu i efikasnu komunikaciju.
  - ▶ **Message Queues:** Servisi komuniciraju asinkrono koristeći redove poruka (npr. RabbitMQ, Kafka).
- ▶ Važno je definisati jasne interfejsa za komunikaciju između servisa.
- ▶ Implementacija sistema za otkrivanje servisa (Service Discovery) može pomoći u upravljanju i otkrivanju dostupnih servisa.

# Primeri upotrebe više servisa

## ▶ **E-commerce aplikacija:**

- ▶ Servis za upravljanje korisnicima (User Service).
- ▶ Servis za upravljanje proizvodima (Product Service).
- ▶ Servis za obradu narudžbina (Order Service).
- ▶ Servis za plaćanja (Payment Service).

## ▶ **Sistem za upravljanje sadržajem (CMS):**

- ▶ Servis za upravljanje člancima (Article Service).
- ▶ Servis za upravljanje korisnicima (User Service).
- ▶ Servis za pretragu (Search Service).
- ▶ Servis za analitiku (Analytics Service).

# Izazovi u korišćenju više servisa

- ▶ **Kompleksnost:** Veći broj servisa povećava kompleksnost sistema.
- ▶ **Upravljanje podacima:** Potreba za sinhronizacijom i konzistentnošću podataka između servisa.
- ▶ **Sigurnost:** Osiguranje sigurnosti komunikacije između servisa.
- ▶ **Nadzor i praćenje:** Potreba za alatom za nadzor i praćenje performansi servisa.

# Alati i tehnologije za rad sa više servisa

- ▶ **Docker:** Omogućava kontejnerizaciju servisa za lakše upravljanje i implementaciju.
- ▶ **Kubernetes:** Orkestracija kontejnera za automatsko skaliranje, balansiranje opterećenja i upravljanje servisima.
- ▶ **API Gateway:** Centralna tačka za upravljanje API pozivima i sigurnošću (npr. Kong, NGINX).
- ▶ **Service Mesh:** Upravljanje komunikacijom između servisa (npr. Istio, Linkerd).



# Zaključak

- ▶ Korišćenje više servisa istovremeno može značajno poboljšati skalabilnost i fleksibilnost sistema.
- ▶ Važno je pažljivo planirati arhitekturu i komunikaciju između servisa.
- ▶ Alati kao što su Docker, Kubernetes i API Gateway olakšavaju rad sa mikrouslužnom arhitekturom.
- ▶ Uprkos izazovima, prednosti mikrouslužne arhitekture čine je popularnim izborom za moderne aplikacije.