Sigurnost operacijskih sustava i aplikacija

Sigurnost mikroservisne arhitekture

Nikola Kušen, 11. 4. 2025.

Pregled predavanja

- Pitanja za ispite
- Motivacija
- Usporedba monolitne i mikroservisne arhitekture
- Slabosti mikroservisne arhitekture
- Dobre prakse
- Zaključak

Pitanja za ispite

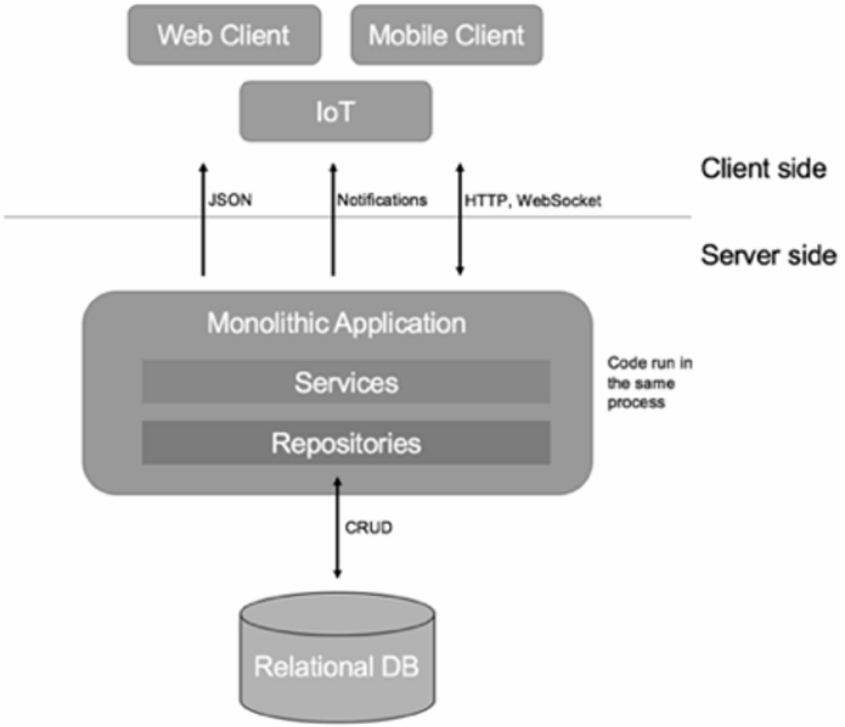
- Zašto mikroservisi imaju veće sigurnosne izazove od monolita?
- Objasnite što je *API Gateway* i navedite njegov značaj u sigurnosti
- Navedite neke metode autentifikacije u mikroservisnoj arhitekturi
- Objasnite princip najmanjih privilegija
- Objasnite kako se sigurnost uklapa u DevSecOps

Motivacija

- Mikroservisna arhitektura je, kao alternativa monolitne, sveprisutna u današnjem tehnološkom svijetu
- Većina organizacija koristi mikroservisnu arhitekturu [3]
  - 85% velikih organizacija (5000+ zaposlenih)
  - 14% planira uvesti
- Vjerojatno ćemo se tijekom karijere susresti s mikroservisima
- Zbog sveprisutnosti arhitekture potrebno je proučiti njene sigurnosne nedostatke i načine na koje se oni mogu riješiti / ublažiti

Tradicionalna monolitna arhitektura

- Kompleksna aplikacija može biti razlomljena na više modula, ali se sve zajedno pušta u pogon
- Ako negdje dođe do greške, cijela aplikacija prestaje raditi
- Skaliranje na razini cijelog procesa
- Promjena jednog dijela zahtjeva kompletnu izgradnju i puštanje u pogon



Slika 1. trorazinska monolitna aplikacija [1]

Mikroservisna arhitektura

- Moduli su neovisni i vrte se u zasebnim kontejnerima
- Definirana sučelja preko kojih komuniciraju
- Na koliko male dijelove treba podijeliti da se smatra mikroservisima?
  - Amazon – 6 do 10 ljudi za jedan servis, nekoliko tisuća linija koda [10]
- Neovisno skaliranje, ažuriranje, verzioniranje i puštanje u pogon

Mikroservisna arhitektura

- Svaki mikroservis može koristiti različite tehnologije
  - Odabir po prednostima tehnologije ili iskustvu razvojnog tima
- Greška u jednom mikroservisu ne uzrokuje pad ostalih, ali ispadom treba rukovati
- Specijalizacija razvojnog tima u užoj domeni
  - Rezultat je veća stručnost

Glavne sigurnosne prijetnje u mikroservisima

- Povećana površina napada zbog distribuirane prirode sustava – više vanjskih sučelja
- Autentifikacija između servisa
- Komunikacija među servisima može biti ranjiva na presretanje podataka
- Teža analiza zapisa (logs) i detekcija anomalija
- Napadi se mogu dogoditi na različitim razinama
    - Mrežnoj, servisnoj, kontejnerskoj, orkestracijskoj i razini otkrivanja servisa

Mrežni napadi

- Mikorservisi su ranjiviji na navedene napade zbog više ulaznih točaka i više komunikacije preko mreže
  - Lakše presresti nego međuprocesnu komunikaciju
- Napad uskraćivanja usluge
- Čovjek u sredini
- Lažiranje ARP adresa
- Lažiranje identiteta
- Prisluškivanje
- TLS napadi

Servisni napadi i ranjivosti

- Loša kontrola pristupa
  - Mikroservisi ranjiviji ako svaki provodi kontrolu, ali može centralizirano u *API Gateway*
- Napad umetanjem SQL-a
  - Svaki servis treba svoju zaštitu, ali šteta je lokalizirana na jedan servis
- Cross-Site Scripting (XSS)
  - Ako je jedan frontend onda isto kao i kod monolita, ali postoji mikrofrontend
- Loše sigurnosne postavke
- Loša autentifikacija i autorizacija
- Nedovoljno zapisivanje i nadzor
- Server side request forgery (SSRF)
  - Veća ranjivost mikroservisa jer su interni servisi izloženi, ne samo vanjski

Kontejnerski napadi

- Udaljeno izvršavanje koda (RCE)
  - Mikroservisi manje ranjivi jer je šteta samo u jednom kontejneru ako je dobro izoliran
- Neovlašteni pristup
  - Ako svaki provodi svoju autorizaciju propusti su češći
- Zloćudni kod u kontejnerima
  - Kod mikroservisa veći broj kontejnera, timova i slika pa je veća šansa da se od nekud provuče zlonamjerni kod

Napadi na orkestracijski sloj

- Orkestracijska komponenta automatski upravlja raspoređivanjem, skaliranjem, povezivanjem i nadzorom kontejnera
- Kubernetes je najpopularniji primjer
- Mikroservisi su ranjiviji zbog kompleksnosti orkestracijskih platformi, monoliti eventualno koriste jednostavne orkestratore
- Kompromitiranje kontrolne ploče
- Lažno predstavljanje servisa
- Krađa vjerodajnica
- Manipulacija skaliranjem

Napadi na razini otkrivanja servisa

- Mikroservisi su ranjiviji jer se DNS može zloupotrijebiti, a monoliti niti nemaju potrebu za automatskim otkrivanjem servisa
- Lažno predstavljanje pri otkrivanju servisa
  - Napadač se predstavlja kao servis, ali ne mora biti u registryju
- Otmica servisa
  - Napadač zamijeni legitimnu registraciju
- Manipulacija registracijom servisa
  - Širi pojam — odnosi se na bilo kakvu zlouporabu sustava registracije servisa

Što je potrebno

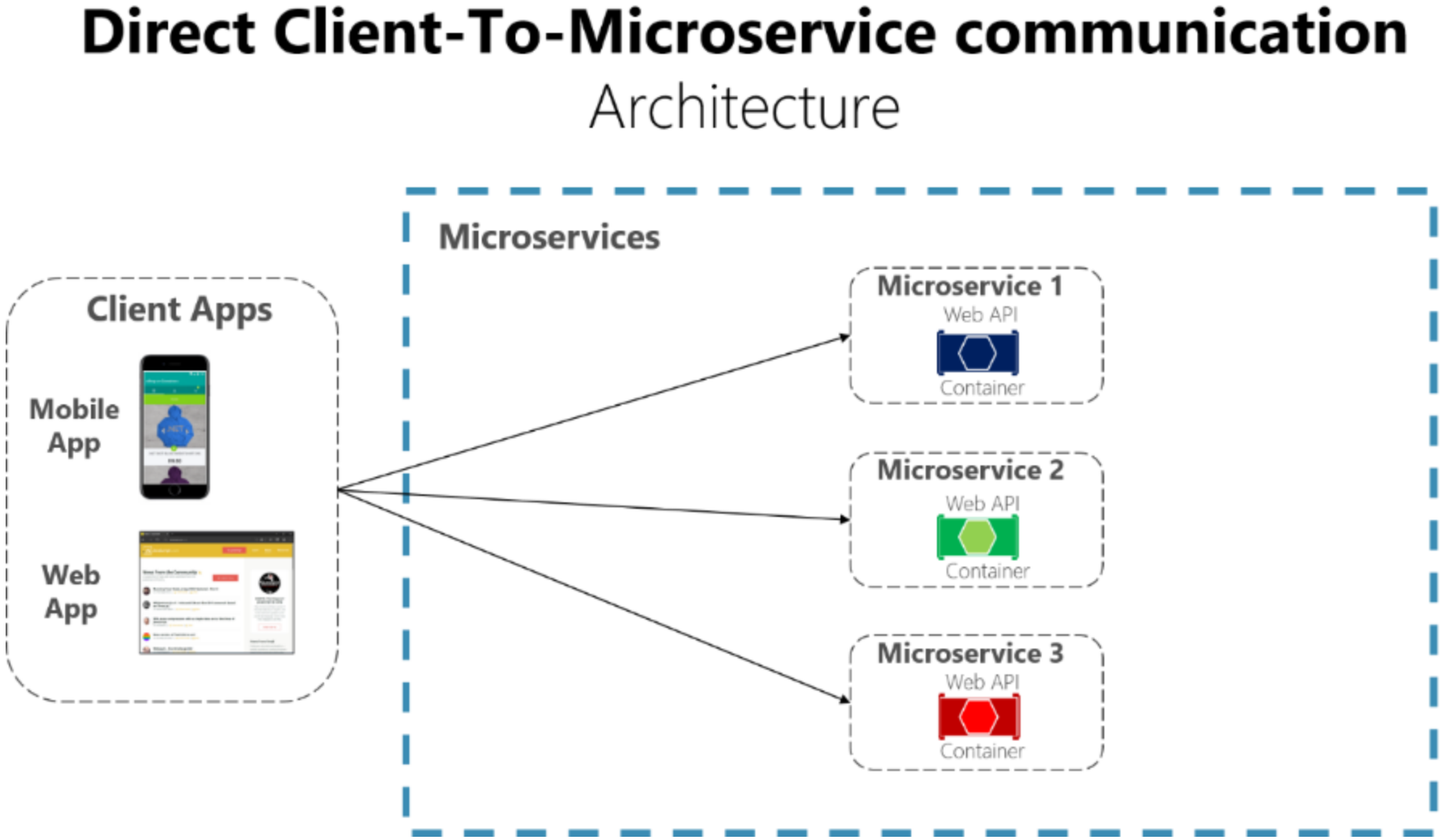
- Koristiti minimalne privilegije i najnižu moguću granicu povjerenja
- Identificirati, minimizirati i ojačati površinu napada
- Osigurati da arhitekti i programeri strogo upravljaju razinama povjerenja
- Podaci moraju biti zaštićeni ne samo na logičkoj razini unutar svakog mikroservisa, već i prilikom prijenosa između servisa

Autentifikacija i autorizacija

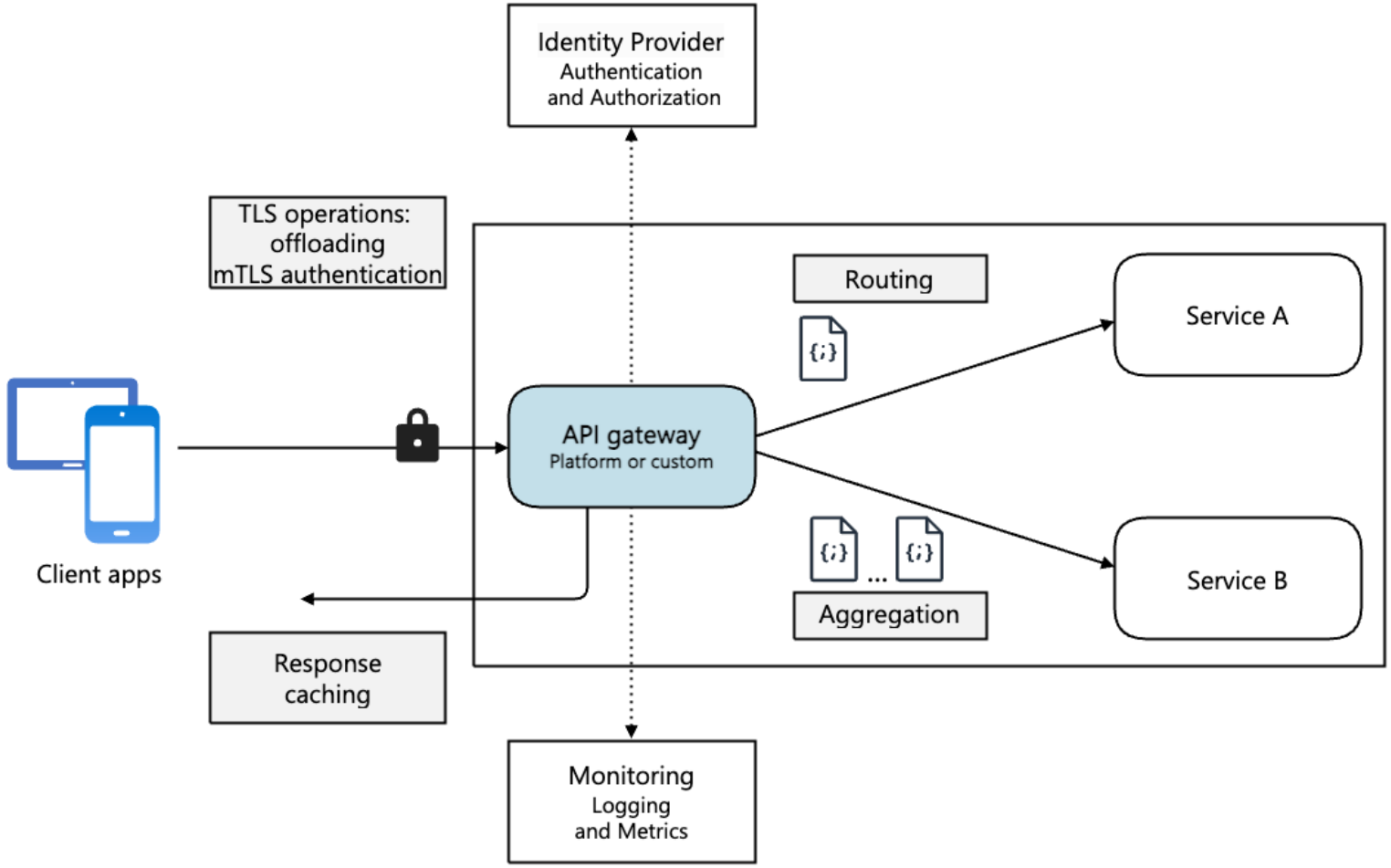
- Autentifikacija – je li korisnik zaista onaj kojim se predstavlja
- Autorizacija – smije li on vršiti određenu akciju
- OAuth 2.0 i OpenID Connect kao standardi za autentifikaciju
- JWT (JSON Web Token) za sigurno upravljanje sesijama
- RBAC (Role-Based Access Control) i ABAC (Attribute-Based Access Control)
- Privilegirani pristup treba biti strogo ograničen
- Princip najmanjih privilegija
  - Korisnici ili procesi trebaju dobiti minimalne privilegije potrebne za obavljanje svojih zadataka

Sigurna komunikacija među mikroservisima

- TLS šifriranje za zaštitu podataka u prijenosu
- Mutual TLS (mTLS) za autentifikaciju servisa
  - Obje strane šalju i provjeravaju certifikat, koristi se u Zero Trust modelima
- API Gateway za komunikaciju s vanjskim svijetom, Service Mesh (npr. Istio) za upravljanje prometom i sigurnošću unutar aplikacije – grubozrnata sigurnost na razini servisa
- Implementacija sigurne razmjene poruka (npr. AMQP, gRPC) – finozrnata sigurnost na razini poruke


Direct Client-To-Microservice communication
Architecture
Microservices
Client Apps
Mobile App
Web App
Microservice 1
Web API
Container
Microservice 2
Web API
Container
Microservice 3
Web API
Container

Slika 2


Identity Provider
Authentication
and Authorization
TLS operations:
offloading
mTLS authentication
Routing
Service A
API gateway
Platform or custom
Client apps
Service B
Aggregation
Response
caching
Monitoring
Logging
and Metrics
Microsoft
Azure

Slika 3

API Gateway

- Slika 2 [6] prikazuje sustav direktne komunikacije klijenata i mikroservisa, a slika 3 [7] sustav s *API Gatewayem*, jedinom točkom ulaska u sustav
- Smanjuje ranjivost na napade jer centralizira proces autentifikacije, validira zahtjeve i ograničava njihov broj
- Skriva unutarnju arhitekturu
- Integracija s WAF (Web Application Firewall) za dodatnu zaštitu

Upravljanje tajnama i konfiguracijom

- Sigurno pohranjivanje tajni pomoću alata poput HashiCorp Vault, AWS Secrets Manager
- Izbjegavanje hardkodiranih lozinki i API ključeva u kodu
- Kontrola pristupa tajnama pomoću strogo definiranih politika
- Redovita promjena ključeva i certifikata
  - Prethodno navedeni alati ovo omogućuju na jednostavan način

Sigurnost podataka i pohrane

- Šifriranje podataka u mirovanju (AES-256)
- Pričuvna pohrana i planovi oporavka u slučaju nesreće
- Pravilan dizajn baze podataka radi smanjenja izloženosti osjetljivim informacijama
  - Odvojeni korisnici za svaki servis s točno određenim pravima
  - Zbog regulatornih zahtjeva (GDPR) najčešće potrebno staviti te informacije u posebnu bazu i kriptirati osjetljiva polja
- Kontrola pristupa bazama podataka kroz granularne dozvole
  - Samo neke dozvole (CRUD) nad određenim tablicama, stupcima ili redovima

Monitoring i detekcija prijetnji

- Centralizirano bilježenje (logging) i analiza podataka pomoću alata poput ELK stacka, Prometheusa
- Intrusion Detection Systems (IDS) za otkrivanje sumnjivih aktivnosti
  - Potpisi poznatih napada, nepoznata računala, strojno učenje
- Automatizirani odgovori na sigurnosne incidente
- Implementacija SIEM (Security Information and Event Management) sustava
  - Prati zapise (logs) te ponašanje korisnika

DevSecOps – Sigurnost u CI/CD okruženju

- Integracija sigurnosnih testova u CI/CD pipeline (SAST, DAST, SCA)
- Skeniranje kontejnera za ranjivosti prije produkcije
- Automatizacija sigurnosnih provjera u svakom koraku razvoja
- Korištenje IaC (Infrastructure as Code) za sigurnu konfiguraciju
  - Sve ide kroz kôd, nije moguće otvoriti port klikom na gumb
  - Lakše pratiti promjene, a postoje i alati koji automatski provjeravaju sigurnost konfigurirane infrastrukture (OPA, Checkov, Terraform Sentinel)

Sigurnost kontejnera i orkestracije

- Kubernetes sigurnosne prakse: RBAC, kapsule (pods) u zaštićenoj okolini, mrežne politike
- Redovito ažuriranje kontejnerskih slika i korištenje minimalnih baza
- Skeniranje kontejnera za ranjivosti (npr. Trivy, Clair)
- Ograničavanje dozvola unutar kontejnera kako bi se smanjio rizik napada

Zaključak

- Alati su dostupni i treba ih koristiti kad je moguće
- Pratiti dobre prakse i sigurnosna istraživanja
- Implementacija Zero Trust modela i sigurnosne provjere u svim fazama razvoja (DevSecOps)
- Kontinuirano praćenje i analiza prijetnji
- Redovito ažuriranje sustava i biblioteka
- Edukacija i podizanje svijesti o sigurnosti unutar tima, rokovi i pritisak izvana ne smiju ugroziti sigurnost

Literatura

- [1] Mateus-Coelho, N., Cruz-Cunha, M., & Ferreira, L. G. (2021). Security in microservices architectures. *Procedia Computer Science*, *181*, 1225–1236.
- [2] Rezaei Nasab, A., Shahin, M., Hoseyni Raviz, S. A., Liang, P., Mashmool, A., & Lenarduzzi, V. (2023). An empirical study of security practices for microservices systems. *Journal of Systems and Software*, *198*, 111563.
- [3] Statista: Do you utilize microservices within your organization? https://www.statista.com/statistics/1236823/microservices-usage-per-organization-size/
- [4] O'Reilly: O'Reilly's Microservices Adoption in 2020 Report Finds that 92% of Organizations are Experiencing Success with Microservices https://www.oreilly.com/pub/pr/3307

Literatura

- [5] Abhishek Kumar: Different types of attacks that can be done on a microservice https://medium.com/@kumarabhishek0388/different-types-of-attacks-that-can-be-done-on-a-microservice-dc312b8a352c
- [6] Microsoft: The API gateway pattern versus the Direct client-to-microservice communication https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern
- [7] Microsoft: Use API gateways in microservices https://learn.microsoft.com/en-us/azure/architecture/microservices/design/gateway
- [8] IBM: What is security information and event management (SIEM)? https://www.ibm.com/think/topics/siem

Literatura

- [9] Fortinet: What Is An Intrusion Detection System (IDS)? https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system#:~:text=IDS%20vs%20IPS-,What%20Is%20An%20Intrusion%20Detection%20System%20(IDS)%3F,any%20security%20risks%20and%20threats
- [10] Cristian Satnic: Amazon, microservices and the birth of AWS cloud computing https://www.linkedin.com/pulse/amazon-microservices-birth-aws-cloud-computing-cristian-satnic/

Dodatna literatura

- Diego Gongora: What is Microfrontend and Why You Should Know About It https://medium.com/@dgongoragamboa/what-is-microfrontend-and-why-you-should-know-about-it-36e812ec9ecc

Hvala!