

Nastínění současných trendů ve vývoji software

AWS serverless služby

Miroslav Hlávka

2021



Obsah

1. Obecně o cloud službách
2. Teamové práce s kódem (git)
3. Představení AWS platformy
4. Designu software běžící na AWS
5. Ukázka využití AWS

Proč povídání vzniklo

- ▶ Osobní preference
- ▶ IT je velmi dynamický obor
- ▶ SŠ a VŠ jsou schopné pokrýt pouze malou část běžně používaných přístupů
- ▶ Vývoj software s sebou nese potřebu stále se vzdělávat
- ▶ Znalost syntaxe 2-3 jazyků opravdu nestačí

Trocha historie

- ▶ Cloudové služby SaaS (software as a service) - gmail(2004), seznam-email (1998)
- ▶ Rozšíření virtualizace s VMware ESX (2001) a VPS (virtual private server) RoseHosting(2001)
- ▶ Na trhu se objevují velcí hráči AWS (Amazon web services, 2006), GCP (Google cloud platform, 2007)
- ▶ Kontejnery jsou dalším krokem k filosofii microservices - FreeBSD(2000), LXC(2008), Docker(2013)
- ▶ Z Google projektu Borg vychází Kubernetes v1.0 (2015)

Další zdroje

<https://www.dailyhostnews.com/the-history-of-web-hosting>

<https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>

Typy cloud služeb

- ▶ IaaS (infrastructure as a service) - server, networking, firewalling, ...
- ▶ PaaS (platform as a service) - vývojové prostředí s OS a dalšími nástroji (např. hostované Kubernetes)
- ▶ SaaS (software as a service) - cloud based aplikace
- ▶ FaaS (function as a service, serverless) - poskytují runtime a vývojář neřeší nic než vývoj

Další zdroje

<https://www.esds.co.in/blog/cloud-computing-types-cloud/>

Výhody FaaS

- ▶ FaaS je developer-centric a vývojáři oblíbená
- ▶ Žádná vazba na Operations a vše je v rukách vývojáře
- ▶ Jedná se o managed services, tedy update OS je na poskytovateli
- ▶ Přístup pay-as-you-use a často o dost levnější než u on-premise řešení
- ▶ Představuje přenesení capex (capital expenditures) do opex (operating expenditures)
- ▶ U startupů a malých projektů mohou být hostingové služby zcela zdarma

Nevýhody FaaS

- ▶ Vendor-lock
- ▶ Kalkulace ceny hostingu nemusí být vždy jednoduchá
- ▶ Poskytovatelé často podporují omezený počet jazyků
- ▶ FaaS mají startup time, tedy odezva může mít zpoždění až desítky sekund

Co budeme vyvíjet

- ▶ Klient požaduje jednoduchý účetní systém
- ▶ Budeme implementovat pouze účetní deník (vkládání, výpis)
- ▶ Front-end bude psán s využitím React a hostován na AWS Amplify
- ▶ Back-end bude v Go a využijeme AWS Lambda
- ▶ Data budou uložena v AWS DynamoDB (NoSQL)

Proč právě AWS

- ▶ Tradice, spolehlivost, příznivá cena (free tier)
- ▶ Stejně služby najdeme u GCP (Firebase, Functions, Firestore) a dalších
- ▶ Většinou neexistuje silný důvod preferovat jednoho či druhého poskytovatele