

Brezstrežniški zaledni sistemi



AWS Lambda



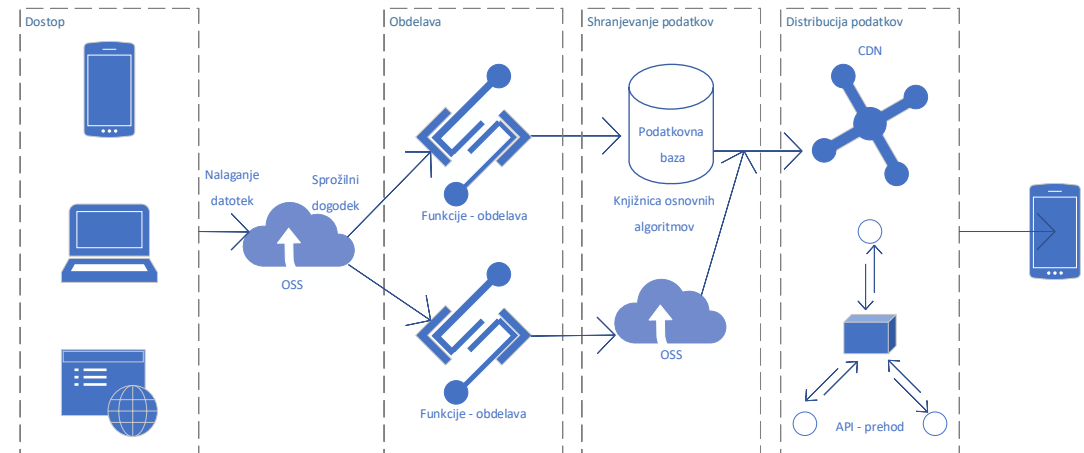
Google Cloud Functions



Azure Functions

Kaj so brezstrežniški zaledni sistemi (serverless)?

- Brezstrežniški zaledni sistem ali “**serverless**” je ideja, da lahko imamo strežniško aplikacijo, pri kateri ne rabimo skrbeti za upravljanje strežnika
- Brezstrežniško računalništvo **ne pomeni**, da za gostovanje in izvajanje kode ne uporabljamo strežnikov
- **IDEJA:**
 - Uporabnikom brezstrežniških zalednih sistemov ni potrebno nameniti časa in sredstev za pripravo strežnikov, vzdrževanje, posodobitve, razširjanje, načrtovanje zmogljivosti...
- Razvijalci se naj osredotočajo na pisanje poslovne logike svoje aplikacije



Kaj so brezstrežniški zaledni sistemi (serverless)?

- Veliko različnih definicij!
- Pogosto slišimo, da so brezstrežniški zaledni sistemi funkcije...

Serverless ~~=~~ Functions

- Funkcije, ki se izvedejo na zahtevo so samo **del** brezstrežniškega zalednega sistema
- Zahteve brezstrežniškega računalništva:
 - 1. Ne skrbimo za upravljanje strežnikov ali procesov
 - 2. Samodejno prilagajanje obsega in samodejno zagotavljanje na podlagi posojila
 - 3. Stroški glede na uporabo
 - 4. Zmožljivosti, ki niso definirane na podlagi velikosti in števila
 - 5. Implicitno visoka razpoložljivost

Serverless – kratka zgodovina

- Google-ov App Engine je bil eden izmed prvih primerov brezstrežniškega računalništva
 - Izdan leta 2008 – izdan “prehitro”
 - Imel preveč omejitev
 - Iz beta verzije šele leta 2011
- Pravi začetki zgodovine brezstrežniškega računalništva se začnejo leta 2012 – članek Ken-a Fromm-a
 - Napoved, da se bodo spletne aplikacije odmaknile od monolitnih vzorcev proti zasnovi distribuiranih sistemov z nizko sklopljenimi komponentami
- 2014 – izid AWS (Amazon Web Services) Lambda – zametek prvega brezstrežniškega računalništva



AWS Lambda

Brezstrežniški zaledni sistemi (serverless) – sestavni deli

- Sestavni deli brezstrežniških zalednih sistemov:
 - Functions As A Service – **FAAS**
 - Backend/Database As A Service – **BAAS**
 - Shramba – **Storage**
 - Pošiljanje sporočil – **Messaging**
 - Varnost – **Security**
 - ...
- Kateri koli izdelek, osredotočen na razvijalce, ki povzema osnovne strežnike === brezstrežniški zaledni sistem
- **Razvijalcem prijazna abstrakcija** – že od nekdaj uporabljamo čim več abstrakcije za lažje opravljanje dela

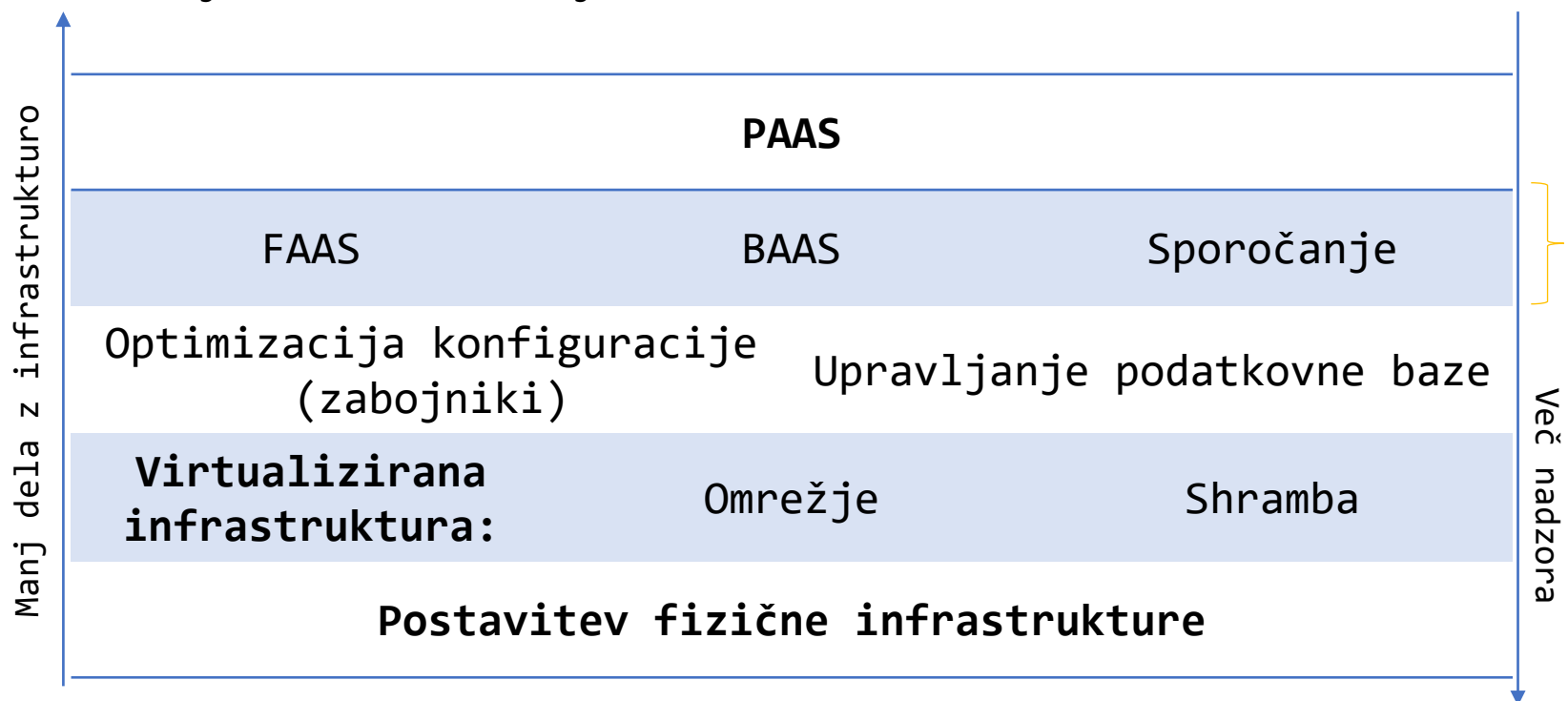
Zakaj uporabiti brezstrežniški zaledni sistemi (serverless) – prednosti



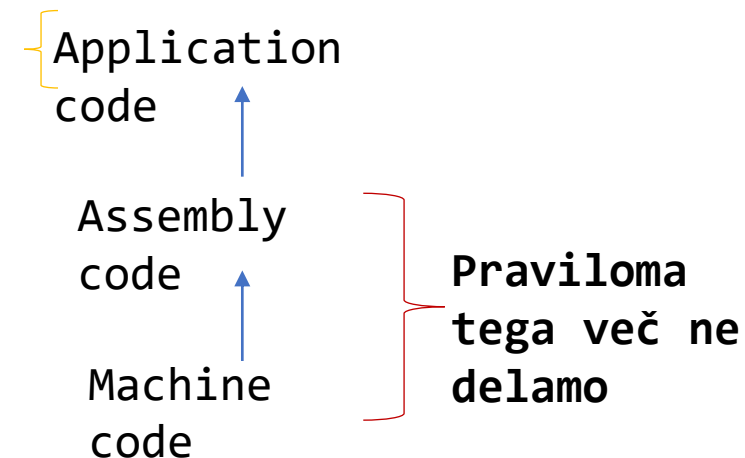
- Smo že slišali za Amazon?
- Amazon odločil, da se bo začel organizirati okoli majhnih avtonomnih skupin
- Zamisel je bila, da je manjšim ekipam lažje sodelovati pri krajših iteracijah in oblikovanju novih idej
- Ko so poskušali zmanjšati količino usklajevanja med dvema ekipama, so ekipe spodbujali k avtonomnemu delu
- Vsaka ekipa je morala skrbeti za načrtovanje, gradnjo in testiranje programske opreme – **pri čemer so skrbeli za vso potrebno infrastrukturo v ozadju**
- Vse skupaj pripeljalo do problema, da so ekipa večino časa namenjale delu, ki sploh ni bilo njihovo delo
- Amazon nastalo težavo rešil na način, da je **izdelal orodja**, ki so njihovim inženirjem omogočila lažje delo (npr. pri infrastrukturi)
- Z novimi orodji razvijalci niso več skrbeli za takšne stvari – **poženemo vrstico kode in vse se izvaja v ozadju**

Zakaj uporabiti brezstrežniški zaledni sistemi (serverless) – prednosti

- Abstrakcija nas naredi bolj produktivne
 - Celotna ideja oblačnih storitev
- Brezstrežniški zaledni sistemi nam omogočajo, da več časa namenimo razvoju kot administraciji



Področje ukvarjanja



Serverless

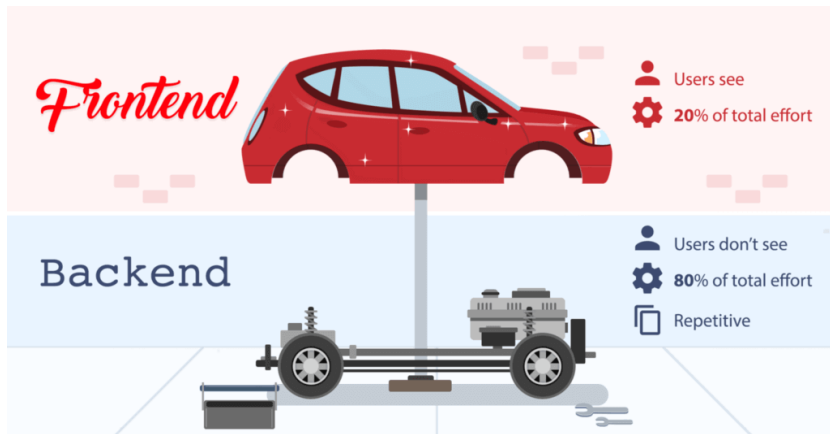


Brezstrežniški zaledni sistemi (serverless) – BAAS - primerjava



- **Zaledni sistem po meri:**
 - Začnemo iz “nule”
 - Skrbimo za infrastrukturo
 - Fleksibilnost
 - Možnost “bolj natančnega” prilagajanja svojim potrebam
 - Visoki stroški razvoja
 - Daljši čas preden pridemo na trg
- **BaaS:**
 - Ponuja v naprej pripravljene “gradnike” in orodja za ustvarjanje kode
 - Glavna prednost je hitrejši razvoj
 - Hitrejši čas, da pridemo na trg
 - Manj možnosti prilagajanja
 - Slediti moramo standardnim arhitekturam

Brezstrežniški zaledni sistemi (serverless) – BAAS - prednosti



- Zakaj?
 - Rešimo dve glavni težavi:
 - Upravljanje in razširjanje oblačne infrastrukture
 - Hitrost razvoja zaledja
- Poslovni razlogi:
 - Cenejše vzdrževanje
 - Prelaganje odgovornosti upravljanja na ponudnika
 - Manj razvijalcev
 - Hitrejši čas prihoda rešitve na trg
- Tehnični razlogi:
 - Osredotočeni smo na razvoj uporabniškega vmesnika
 - Manj “boilerplate” kode
 - Standardiziranje okolja za razvoj
 - Razvijalci zalednih sistemov lahko pišejo samo pomembne vrstice kode
 - Zagotavlja funkcije, pripravljene za uporabo, kot so preverjanje pristnosti, shranjevanje podatkov in iskanje
 - Lažje podvajanje aplikacij in poganjanje testnih okolij
 - Ustrezne varnostne nastavitve in postopki varnostnega kopiranja



Firebase



K U Z Z L E



AWS Amplify



Progress® Kinvey®



Back4App



Azure



Parse

Brezstrežniški
zaledni sistemi
(serverless) –
BAAS –
ponudniki