

DIZAJN SERVERLESS WEB APLIKACIJA NA AMAZON PLATFORMI DESIGN OF SERVERLESS WEB APPLICATIONS USING AMAZON SERVICES

Aleksandar Ratkov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisan je razvoj serverless web aplikacije za vođenje evidencije o godišnjim odmorima zaposlenih upotrebom Serverless radnog okvira i AWS (Amazon Web Services) servisa.

Ključne reči: AWS, Serverless, Cloud

Abstract – This paper presents an implementation of serverless web application for managing vacation days of employees using Serverless framework and services which are provided by AWS (Amazon Web Services).

Keywords: AWS, Serverless, Cloud

1. UVOD

Najnovija dostignuća u tehnologiji svakodnevno vrše pritisak na softverske programere u cilju brže implementacije novih funkcionalnosti i samom generisanju poslovne vrednosti projekta. Vreme koje je potrebno projektu da pređe svoj put od razvoja do tržišta postala je jako bitna metrika, kako zbog finansija tako i zbog straha od konkurencije. U prošlosti programeri su pored razvijanja same aplikacije trošili mnogo vremena na planiranje, implementaciju i održavanje celokupne infrastrukture na kojoj se aplikacija izvršavala.

Pojavom virtualizacije rešen je problem neiskorišćenosti hardverskih resursa što je rezultovalo značajnim smanjenjem troškova. Ovakav splet okolnosti doveo je do uspona cloud computinga, gde programeri počinju da tretiraju hardver kao uslužni program. Napredak virtualizacije bazirane na kontejnerima učinio je da se programeri postepeno prebacuju sa velikih monolitnih arhitektura na arhitekturu zasnovanu na distribuiranim mikroservisima. Ovakav iskorak prema kontejnerima i arhitekturi mikroservisa iznedrio je novu paradigmu koja je poznata kao serverless ili računarstvo bez servera.

2. CLOUD COMPUTING

Tradicionalne poslovne aplikacije su oduvek bile komplikovane i veoma skupe. Pored neophodnog hardvera i licenci za njihovo pokretanje, neophodan je i tim stručnjaka koji će voditi računa o konfiguraciji, testiranju, bezbednosti i ažuriranju istih.

Ovi problemi rešeni su upotrebom cloud computing tehnologije koja akcenat stavlja na obavljanje poslovne logike, a kreiranje i održavanje celokupne arhitekture

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

sistema zadatak je cloud provajdera. Postoji tri osnovna tipa cloud computinga:

- SaaS (Software as a Service)
- PaaS (Platform as a Service)
- IaaS (Infrastructure as a Service)

Softver kao usluga predstavlja model isporuke softvera na osnovu zahteva prema kojem korisnici plaćaju pristup i korišćenje funkcionalnosti softvera u obliku web aplikacija, preko Interneta. Platforma kao usluga pruža usluge neophodne za podršku celokupnog životnog ciklusa programskog proizvoda, od samog dizajniranja pa sve do isporuke aplikacija krajnjim korisnicima. Infrastruktura kao usluga predstavlja koncept iznajmljivanja hardverskih resursa kao usluge korisniku kojem je pružena mogućnost korišćenja celokupne računarske infrastrukture, uglavnom kao virtualne platforme.

2.1. Serverless Computing

Serverless computing ili jednom rečju Serverless je relativno novi koncept u softverskoj arhitekturi, koji je izazvao veliku pažnju u zadnjih par godina zahvaljujući velikom broju knjiga, projekata otvorenog koda, konferencija na ovu temu kao i ogromnim ulaganjima vodeće trojke cloud providera - Amazona, Microsofta i Googla. Najkraće rečeno, serverless arhitektura je način razvoja i pokretanja aplikacija i servisa bez potrebe za održavanjem celokupne infrastrukture. Iako sam naziv može pogrešno da sugeriše, u pitanju nije pokretanje aplikacija bez servera. Pored toga što serverless izdvaja pozadinsku infrastrukturu dalje od programera, serveri su i dalje uključeni u izvršavanje same aplikacije. Naziv serverless se koristi zbog toga što podešavanja i održavanje samog servera nije zadatak osobe koja se bavi kreiranjem sistema ili aplikacije, već je to odgovornost cloud providera. U okviru serverless-a postoje dva tipa arhitekture koja se međusobno prepliću i to su:

- BaaS (*Backend as a Service*)
- FaaS (*Function as a Service*)

Backend kao usluga je servisni model u cloud computingu koji služi kao posrednik koji pruža programerima načine da povežu svoje web i mobilne aplikacije sa cloud uslugama preko aplikacionih programerskih interfejsa i softverskih paketa. Korišćenje ovih usluga pored uštede vremena i novca u mnogome olakšavaju i rad programera koji ne moraju samostalno da implementiraju već postojeće funkcionalnosti. Funkcija kao usluga je kategorija usluga u cloud computing-u koja pruža platformu koja omogućava korisnicima da razvijaju, pokreću i upravljaju aplikativnim funkcionalnostima bez

potrebe za izgradnjom i održavanjem celokupne serverske infrastrukture.

3. IMPLEMENTACIJA

U radu je opisana implementacija serverless web aplikacije za vođenje evidencije o godišnjim odmorima zaposlenih. Backend deo aplikacije kreiran je upotrebom Serverless radnog okvira koristeći Amazon Web Services dok je frontend deo implementiran pomoću Vue.js javascript radnog okvira.

3.1. Serverless radni okvir

Serverless radni okvir predstavlja besplatan projekat otvorenog koda koji je napisan pomoću Node.js programskog jezika, a služi za jednostavno kreiranje i deploy-ovanje serverless aplikacija koristeći command-line interface. Velika prednosti ovog radnog okvira je nezavisnost od cloud provajdera i kompatibilnost cloud funkcija u okviru različitih provajdera. Pored toga ne postoje striktna ograničenja po pitanju jezika koji se koristi za pisanje samih funkcija, već je na programeru da izabere neki od ponuđenih jezika poput Node.js, Python, Java, Go, Scala, C#. Sa Serverless radnim okvirom čitava aplikacija je definisana pomoću jednostavnog konfiguracionog fajla u yaml formatu. Jedan od preduslova za upotrebu ovog radnog okvira jeste prethodno instaliran Node.js čija verzija mora biti 6.5.0 ili novija.

3.2. Amazon Web Services

Amazon Web Services je odeljenje Amazon-a koje pruža platforme za cloud computing na zahtev pojedincima i kompanijama na bazi plaćene pretplate. Tehnologija omogućava pretplatnicima da imaju na raspolaganju virtuelnu grupu računara, dostupnu sve vreme, putem Interneta. AWS-ova verzija virtuelnih računara imitira većinu stavki pravog računara, uključujući hardver, izbor operativnih sistema, umrežavanje i unapred učitani aplikativni softver poput web servera, baze podataka itd. Troškovi za usluge korišćenja usluga AWS servisa se zasnivaju na kombinaciji upotrebe, karakteristika hardvera, OS, softvera koje je izabrao pretplatnik.

3.2.1 Cognito

Amazon Cognito je AWS-ov proizvod koji kontroliše autentifikaciju korisnika i pristup mobilnim i web aplikacijama na uređajima povezanim sa internetom. Ova usluga čuva i sinhronizuje podatke krajnjih korisnika, što omogućava programeru da se fokusira na pisanje poslovne logike aplikacije umesto na implementaciju dela koji je zadužen za autentifikaciju korisnika. Dve glavne opcije Amazon Cognita su user pools i identity pools. Prva opcija se odnosi na sakupljanje atributa profila korisnika u direktorijume koji se nazivaju grupe korisnika koje aplikacija koristi za logovanje i registraciju korisnika. Druga opcija, tzv. skup identiteta služi za obezbeđivanje AWS kredencijala korisnicima u cilju kontrolisanja pristupa AWS resursima.

3.2.2 AWS Lambda

Aws Lambda je AWS-ov servis koji omogućuje programiranje funkcija po principu plaćanja na osnovu upotrebe, bez potrebe za obezbeđivanjem memorije i

računarskih resursa za izvršavanje iste. Pisanje koda za Lambda funkciju ne zahteva poznavanje nekog novog programskog jezika već podržava postojeće programske jezike poput Node.js, Python, Java i C #, Go. Lambda funkcije su stateless, što omogućava Lambdi brzo pokretanje onoliko kopija funkcije koliko je potrebno za skaliranje kako bi se obradili svi dolazni događaji. Kao i kod svih FaaS servisa, pa i kod lambde postoje ograničenja po pitanju trajanja izvršavanja same funkcije. Trenutni limit što se tiče izvršavanja AWS Lambda funkcije je 15 minuta, nakon čega dolazi do prekida izvršavanja iste.

3.2.3 API Gateway

AWS API Gateway je servis koji se koristi za kreiranje, objavljivanje, održavanje, nadgledanje i zaštitu REST (Representational state transfer) i WebSocket API-ja. Ovaj AWS-ov servis obezbeđuje kreiranje robusnih, sigurnih i skalabilnih API-ja koji pristupaju AWS-ovim ili drugim web servisima ili podacima sačuvanim na AWS Cloud-u. U radu je API Gateway korišćen za kreiranje REST API-ja koji se sastoji od ruta, a svaka ruta je povezana sa resursom koji rukuje tom rutom. API Gateway se sastoji od funkcija, a svaka funkcija sadrži dva elementa. Prvi element je rukovaoc koji je u ovom slučaju AWS Lambda čije će izvršavanje biti aktivirano određenim događajem. Drugi element je događaj koji aktivira Lambda funkciju, u ovom slučaju HTTP poziv koji je određenog metoda i upućen je ka određenoj ruti. Kontrola pristupa samom API-ju je obezbeđena upotrebom authorizer-a koji može biti kreiran na dva načina. Prvi je upotrebom Cognito user pool-a koji vrši funkciju authorizer-a, a drugi je pisanjem nestandardne Lambda funkcije koja sadržati potrebnu logiku za autorizaciju i autentifikaciju pozivaoca zahteva.

3.2.4 DynamoDB

Amazon DynamoDB predstavlja NoSQL (not only structured query language) bazu podataka koja pruža brze performanse uz besprekornu skalabilnost. DynamoDB baza omogućava uklanjanje administrativnih opterećenja u radu i skaliranju distribuirane baze podataka, te s toga nema potrebe o brizi oko izbora hardvera, podešavanja i konfiguracije, replikaciji, krpiljenju softvera ili skaliranju klastera. Takođe, DinamoDB nudi i enkripciju, čime se eliminiše složenost u zaštiti osetljivih podataka. Uz DynamoDB, moguće je kreiranje tabela baze podataka koje mogu sačuvati i vratiti bilo koju količinu podataka i pored toga opslužiti bilo koji nivo saobraćaja zahteva. DynamoDB pruža mogućnost kreiranja rezervne kopije na zahtev kao i oporavak u trenutku za AWS DynamoDB tabele. Pored ovoga omogućeno je automatsko brisanje stavki koje nisu aktuelne iz tabela kako bi se smanjila upotreba memorije kao i troškovi čuvanja podataka koji nisu više relevantni.

3.2.5 AWS Simple Email Service

Simple Email Service predstavlja AWS-ovu e-mail platformu koja pruža jednostavan i isplativ način za slanje i primanje elektronske pošte koristeći vlastite adrese e-pošte ili domene. AWS SES nudi visoku stopu isporuke e-pošte i jednostavan, trenutni pristup statistici slanja e-pošte upotrebom AWS Management Console. Takođe pruža gotova obaveštenja za uspešne i neuspešne isporuke

e-pošte kao i žalbe putem e-pošte. Amazon SES može slati povratne i žalbene poruke putem e-pošte ili putem AWS SNS (Simple Notification Service) servisa. Amazon SES nudi sandbox okruženje tj. simulator poštanskog sandučeta koji testira kako će aplikacija obraditi različite moguće situacije slanja e-pošte. Još jedna od upotreba simulatora je otkrivanje maksimalne propusnosti sistema bez probijanja dnevnog ograničenja za slanje e-pošte. Slanje e-pošte u ovom radu implementirano je upotrebom AWS Lambda funkcije i AWS SES servisa.

3.2.6 AWS Simple Storage Service

Simple Storage Service (S3) je vodeći servis ovog tipa u industriji koji nudi skalabilnost, dostupnost podataka, sigurnost i performanse kada je u pitanju skladištenje objekata. To znači da korisnici svih veličina i industrija mogu da ga koriste za čuvanje i zaštitu bilo koje količine podataka za različite slučajeve upotrebe, kao što su web sajtovi, mobilne aplikacije, oporavak i vraćanje podataka, arhiviranje, poslovne aplikacije, IoT uređaji i analitika velikih količina podataka.

Amazon S3 pruža jednostavne funkcije upravljanja, što omogućava organizaciju podataka kao i podešavanje kontrole pristupa kako bi zadovoljile specifične poslovne i organizacione zahteve. Amazon tvrdi da je S3 dizajniran sa skoro stoprocentom izdržljivošću te ne čudi podatak da se na S3 čuvaju podaci za milione aplikacija za kompanije širom sveta. S3 automatski kreira i skladišti kopije svih objekata na više sistema, što omogućava dostupnost podataka kad god je to potrebno, a sami podaci su zaštićeni od kvarova, grešaka i pretnji. Da bi se objekti skladištili na Amazon S3 neophodno je kreirati bucket. Imena S3 bucket-a se specificiraju prilikom njegovog kreiranja i globalno su jedinstvena, bez obzira na AWS region u kojem je bucket kreiran.

AWS S3 podržava različite opcije za konfigurisanje bucket-a. Moguće je konfigurisanje bucket-a za hosting web sajta, zatim konfigurisanje kako bi se upravljalo životnim ciklusom objekata u bucket-u kao i konfigurisanje bucket-a da loguje sve pristupe ka bucket-u. S3 nudi kontrolu pristupa bucket-u upotrebom korisničkih i bucket polisa koje su napisane u JSON (JavaScript Object Notation) formatu. U radu AWS-ov S3 servis je korišćen za hostovanje same web aplikacije, što je zahtevalo definisanje javnog pristupa samom bucket-u.

4. ZAKLJUČAK

Tokom godina serversko računarstvo je evoluiralo preko direktne upotrebe metalnih servera, preko virtuelizacije pa sve do koncepta računarstva bez servera (serverless). Koncept serverless računarstva omogućava brži i ekonomičniji razvoj aplikacija čemu se danas svakako teži.

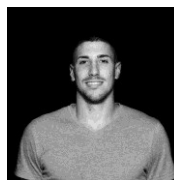
Brži razvoj aplikacije moguć je jer programeri više nemaju obavezu za implementacijom i održavanjem kompletne infrastrukture na kojoj se aplikacija izvršava. Samim tim sa finansijskog aspekta nema potrebe za nabavkom skupocenog hardvera već se koriste usluge cloud provajdera koje se plaćaju isključivo na osnovu njihove upotrebe.

Dalji pravci razvoja ovog rada bi se odnosili na automatizaciju celokupnog procesa razvoja i puštanja u produkciju aplikacije. Težilo bi se kreiranju CI/CD (Continuous Integration/Continuous Delivery) toka. Continuous Integration bi omogućio konstantu integraciju novih funkcionalnosti kao i njihovo testiranje i otkrivanje grešaka pre nego što one dospeju u produkciju. Continuous Delivery bi omogućio kraće cikluse puštanja aplikacije u produkciju koji bi bili automatizovani, što bi svakako rezultovalo smanjenjem troškova, povećanjem kvaliteta koda, srećnijim timovima koji rade na projektu kao i klijentima koji na ovaj način neuporedivo brže dobijaju nove funkcionalnosti u produkciji.

5. LITERATURA

- [1] Cloud Computing, <https://www.salesforce.com/what-is-cloud-computing/>
- [2] Cloud Computing, https://en.wikipedia.org/wiki/Cloud_computing
- [3] Serverless Computing, https://en.wikipedia.org/wiki/Serverless_computing
- [4] Serverless Computing, <https://martinfowler.com/articles/serverless.html>
- [5] Serverless Framework, https://en.wikipedia.org/wiki/Serverless_Framework
- [6] Amazon Web Services, https://en.wikipedia.org/wiki/Amazon_Web_Services
- [7] AWS Cognito, <https://docs.aws.amazon.com/cognito/latest/developer-guide/what-is-amazon-cognito.html>
- [8] AWS Lambda, <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>
- [9] AWS API Gateway, <https://docs.aws.amazon.com/apigateway/latest/developer-guide/welcome.html>
- [9] Serverless Computing, <https://martinfowler.com/articles/serverless.html>
- [10] AWS DynamoDB, <https://docs.aws.amazon.com/amazondynamodb/latest/developer-guide/Introduction.html>
- [11] AWS SES, <https://docs.aws.amazon.com/ses/latest/DeveloperGuide/Welcome.html>
- [12] AWS S3, <https://searchaws.techtarget.com/definition/Amazon-Simple-Storage-Service-Amazon-S3>

Kratka biografija:



Aleksandar Ratkov rođen je u Beogradu 1992. god. Osnovnu školu „Boško Pal-kovljević Pinki” završio je 2007. godine. Gimnaziju „Branko Radičević ” u Staroj Pazovi završio je 2011. godine. Zvanje diplomirani inženjer elektrotehnike i računarstva stekao je 2015. godine.