Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 4.4
DOI: https://doi.org/10.24867/33BE35Trivunovic

SAMOOBNAVLJAJUĆI KOD NA AWS PLATFORMI

SELF HEALING CODE ON AWS PLATFORM

Dragana Trivunović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad istražuje mogućnost rešenja samoobnavljajućeg koda koji koristi AWS platformu i generativnu veštačku inteligenciju. Mogućnosti koje otvara složeno rešenje koda koji se samoobnavlja se ogleda u smanjivanju utrošenog vremena razvojnog inženjera i brzom detektovanju greške u softveru i njenom otklanjanju. Glavni servis koji je centar arhitekture ovog rešenja je servis generativne veštačke inteligencije koji se koristi za sugestiju, dopunu, ispravku, pojašnjenje, optimizaciju, transformaciju i poboljšanje softverskog rešenja.*

Ključne reči: *Cloud, AWS, Generativni AI*

Abstract – *This paper explores the feasibility of a self-healing code solution utilizing the AWS platform and generative artificial intelligence. The potential of a complex self-healing code solution lies in reducing the time spent by software engineers, enabling quick error detection in the software, and facilitating error correction. The core service at the heart of this architecture is a generative AI service, used for suggesting, completing, correcting, clarifying, optimizing, transforming, and enhancing the software solution.*

Keywords: *Cloud, AWS, Generative AI*

1. UVOD

Era naprednog razvitka veštačke inteligencije dovela je do, nekad nezamislivih, tehničkih rešenja. Primena ovakvih rešenja je u velikoj meri olakšala niz zadataka u razvoju softvera. Optimizacija i automatizacija su ključni faktori koji izdvajaju rešenja sa primenom veštačke inteligencije smanjujući cenu realizacije, utrošeno vreme i potrebne resurse za razvoj softvera. Pomoć koju veštačka inteligencija pruža prilikom razvoja i održavanja softverskih rešenja omogućila je smanjenje vremena koje je potrebno da se pronađe i popravi greška (engl. bug). U trci za napredna rešenja veštačke inteligencije i mašinskog učenja, sa svojim inovativnim pristupom, ističe se AWS (Amazon Web Services) kompanija. Načini korišćenja veštačke inteligencije unutar njihovih proizvoda dovode do poboljšanog iskustva korisnika, povećane produktivnosti zaposlenih, kreativnijeg marketinga i optimizovanih procesa unutar kompanije. Ovaj rad istražuje AWS Self healing kod, analizirajući njegovu arhitekturu, funkcionalnosti i praktičnu primenu.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, red. prof.

Na AWS platformi se mogu koristiti već istrenirani modeli veštačke inteligencije i infrastruktura napravljenih servisa. Osim fokusa na veštačku inteligenciju koji čini samo jedan deo ponuđenih rešenja, na AWS platformi postoji čitav niz softverskih rešenja koji čine infrastrukturu AWS-ovog „oblaka“ (engl. Cloud). Usluge koje se pružaju su skladištenje i arhiviranje podataka, procesuiranje podataka, izgradnja API interfejsa, privatni VPN, elastično balansiranje opterećenja nadolazećeg saobraćaja na veb sajtu, razne analitike i metrike podataka korišćenjem Big Data tehnologija, implementiranja politike sigurnosti, identiteta i saglasnosti, itd.

U ovom radu će biti obrađeni neki od servisa koji su iskorišćeni u stvaranju rešenja za kod koji ima mogućnost samoobnavljanja, detekcije grešaka i podizanja ispravaka koda u izvorni kod.

2. AMAZON WEB SERVICES

U poređenju sa klasičnim računarstvom, AWS kao pionir ideje računarstva u oblaku 2000-ih godina, donosi novosti koje su zauvek izmenile računarstvo. Najveći benefiti se ogledaju u skalabilnosti, gde manje firme imaju mogućnost da plaćaju samo onoliko koliko im u datom trenutku treba, te sa povećanjem obima posla povećaju i obim AWS-ovih usluga. U prošlosti su firme koristile privatne centre podataka sa privatnim hardverom i serverima, koji su u poređenju sa servisima na „oblaku“ bile izuzetno skuplje. Skaliranje na AWS-u može da bude i automatizovano i brzo, te je ovakav vid poslovanja mnogo pogodniji. U današnje vreme sve modernizovane firme i kompanije koriste rešenja računarstva u oblaku [5].

2.1. AWS servisi

AWS takođe nudi svojim korisnicima razvijeni sistem za implementiranje politike sigurnosti, saglasnosti i identiteta. Postoje servisi za nadzor sistema koji povećavaju sigurnost. Štiti svoje korisnike od ispada sistema stavljajući akcenat na pouzdanost svojih usluga. AWS garantuje dostupnost korisničkih aplikacija pružanjem usluga na 80 zona dostupnosti i više od 25 globalnih regija, što ga čini odličnim izborom za kompanije koje posluju globalno. Pruža razvojne alate za softver koji su laki za korišćenje i omogućava brzo podizanje korisničkih aplikacija i veb sajtova. Postoje nekoliko tipova razvojih modela računarstva u oblaku [3].

U modelu infrastrukture kao servisa, korisniku je omogućeno korišćenje računarske

infrastrukture u vidu virtuelne platforme kao što su virtuelne mašine i serveri zaskladištenje i upravljanje podacima. Primeri modela infrastrukture kao servisa u AWS-u su servisi poput Elastic Computing 2, Simple Storage Service – S3, Relational Database Service – RDS.

Softver kao servis je model koji se fokusira na dostavljanje aplikacije korisniku preko interneta. AWS ne pruža gotova Softver kao servis rešenja, ali pruža mogućnost svojim korisnicima da naprave svoje samostalne aplikacije bazirane na ovom modelu.

Platforma kao servis je model koji pruža platformu, alate i razvojna okruženja softverskim inženjerima za razvoj softverskih rešenja. Primer je Content Delivery Network – CDN servis koji predstavlja mrežu povezanih servera koji ubrzavaju učitavanje veb stranica za aplikacije koje su orijentisane na podatke. Takav jedan servis je AWS-ov CloudFront. Platforma kao servis je implementirana kao model u servisima za balansiranje protoka podataka. Elastic Load Balancing – ELB je servis koji omogućava distribuciju opterećenja aplikacije na serverima zarad poboljšanja skalabilnosti aplikacije. Još jedan primer ovog modela se nalazi u Amazon CloudWatch servisu koji služi za monitoring podataka i aplikacije, preko konstrukcije log datoteka i raznih metrika.

Back-end kao servis je model koji pomaže razvojnim inženjerima da se fokusiraju na front-end deo aplikacije. U ovom modelu je serverski deo aplikacije već napravljen i umesto korisnika upravlja bazom podataka, skladištenjem podataka, autentikacijom korisnika, notifikacija na aplikaciji, veb-hostinga, itd. Primer ovog modela na AWS-u je AWS Amplify. Ovaj model omogućava front-end inženjerima da konstruišu full-stack aplikacije.

Funkcija kao servis je model računarstva u oblaku gde je fokus stavljen na pokretanje funkcija kao odgovor na neki događaj, bez potrebe da se konfiguriše kompleksna infrastruktura virtuelnih mašina i upravljanje operativnim sistemima i procesima održavanja veb servera. Sa ovim modelom, veb server se može podeliti na funkcionalnosti koje se mogu automatski skalirati. Sa ovim modelom, plaća se samo kada se funkcionalnost iskoristi, a ne po satu ili količinskom najmu kao kod ostalih modela. Primer ovog modela je AWS Lambda servis. Ovaj servis izvršava kod na visoko dostupnoj infrastrukturi za obradu podataka i vrši administraciju svih resursa za obradu, uključujući i administraciju operativnog sistema i servera. Takođe, ovaj servis automatski skalira i potražuje potrebne resurse za obradu podataka.

3. AWS SELF HEALING CODE

Pod pritiskom brzog razvoja, programeri i menadžeri su često suočeni sa izborom trošenja vremena na popravke grešaka koda koji je već u produkciji, ili ostavljanjem greške i problema u „back log“-u gde se gomila kao tehnički dug. Prvi izbor znači više utrošenog vremena i novca, a drugi izbor dovodi do nezadovoljnih klijenata i lošeg korisničkog iskustva.

Rešenje do kojeg su došli inženjeri iz Amazon AWS kompanije je kod koji ima mogućnost introspekcije na osnovu prijavljenih grešaka na aplikaciji, mogućnost ispravke koda i podizanja te ispravke na izvorni kod. Ključni deo ovog rešenja je generativni model veštačke inteligencije koji nadopunjuje i ispravlja kod.

3.1. ARHITEKTURA REŠENJA

Ovo rešenje je napravljeno kombinovanjem Amazon CloudWatch-a, AWS Lambda i Amazon Bedrock servisa kako bi kreirali sveobuhvatan sistem koji automatski otkriva i popravlja greške radi poboljšavanja pouzdanosti aplikacije i celokupnog korisničkog iskustva. U ovom sistemu, ispravljač grešaka se povezuje sa CloudWatch evidencijom zapisa grešaka aplikacije preko Lambda pretplate. Svi zapisnici koji sadrže greške aplikacije šalju se na obradu, gde Lambda funkcija kreira prompt, uključujući praćenje steka (engl. „stack“) i relevantne datoteke koda, a zatim ga šalje u Amazon Bedrock (Claude v1 model) da generiše ispravke koda. Izmenjeni kod se zatim šalje u kontrolu izvora (Git) i kreira zahtev za povlačenje za pregled i primenu [6].

Ovakvo rešenje pruža niz funkcionalnosti: Automatsko otkrivanje praćenja steka (greške): Implementira pretplate na CloudWatch evidencije za automatsko filtriranje i otkrivanje tragova steka; Praćenje grešaka: Automatski prati stanje obrade grešaka u Amazon DynamoDB; Deduplikacija: Deduplikuje tragove steka da bi se izbegla suvišna obrada; Kreiranje zahteva za povlačenje: Integriše se sa sistemima kontrole izvora za automatsko kreiranje zahteva za povlačenje, koji uključuju ispravke grešaka

Postoji nekoliko servisa koji se koriste u ovom rešenju a koji su spomenuti u prethodnim poglavljima. Amazon CloudWatch Core servis pruža podatke evidencije grešaka za problematičan izvorni kod. AWS Lambda Core servis implementira automatizaciju za brzo generisanje i interakciju sa BedRock-om. Amazon Bedrock Core servis pruža širok skup mogućnosti za izgradnju generativnih AI aplikacija koje analiziraju, popravljaju i vraćaju izmenjeni problematičan izvorni kod. Amazon Simple Queue Service (Amazon SQS) servis pruža grupnu obradu i kontrolu istovremenosti za Lambda funkciju. Amazon DynamoDB Core servis čuva trag steka koda sa evidencijama grešaka i prenosi podatke. Pomoćni servis Amazon Systems Manager čuva tajne parametre u skladištu [12].

3.1.1. Amazon CloudWatch

Amazon CloudWatch je servis koji nadgleda aplikacije, reaguje na promene u performansama, optimizuje korišćenje resursa i pruža uvid u operativno zdravlje. Prikupljanjem podataka preko AWS resursa, CloudWatch daje uvid u performanse celog sistema i omogućava korisnicima da podese alarme, automatski reaguju na promene i steknu jedinstven pogled na operativno zdravlje. CloudWatch je kao servis napravljen po modelu Platforma kao servis.

CloudWatch funkcioniše tako što akumulira metriku i evidenciju iz AWS resursa, analizira te podatke i pruža vizuelne prikaze i upozorenja. Sastoji se od 3 glavne komponente: deo za praćenje i prikupljanje metrike koji prikuplja podatke iz resursa, dugoročno skladištenje tih podataka u CloudWatch-u, kao i vizuelizacije i alarme na osnovu uskladištenih podataka.

3.1.2. AWS Lambda

Moguće je koristiti AWS Lambda za pokretanje koda bez obezbeđivanja ili upravljanja serverima.

Lambda pokreće kod na računarskoj infrastrukturi visoke dostupnosti i obavlja svu administraciju računarskih resursa, uključujući održavanje servera i operativnog sistema, obezbeđivanje kapaciteta i automatsko skaliranje i evidentiranje. Sa Lambda-om, sve što treba da uradite je da unesete svoj kod u jednom od jezika izvođenja koje Lambda podržava [1].

Kod se organizuje u Lambda funkcije. Lambda usluga pokreće funkciju samo kada je to potrebno i automatski se podešava. Plaća se samo vreme koje je utrošeno na pokretanje i izvršavanje funkcije — nema naknade kada kod nije pokrenut.

Lambda je idealna računarska usluga za scenarije aplikacija koje treba brzo da se povećaju i smanje na nulu kada nisu tražene.

Kada se koristi Lambda, odgovornost je na inženjeru samo za njegov kod. Lambda upravlja računarskom flotom koja nudi balans memorije, CPU-a, mreže i drugih resursa za pokretanje vašeg koda. Pošto Lambda upravlja ovim resursima, nije moguće prijaviti se da bi se izračunale instance ili prilagodili operativni sistem na predviđenim vremenima izvođenja. Lambda obavlja operativne i administrativne aktivnosti u korisničko ime, uključujući upravljanje kapacitetom, praćenje i evidentiranje korisničkih Lambda funkcija [2].

3.1.3. Amazon DynamoDB

DynamoDB je bez-serverska, NoSQL, potpuno upravljana baza podataka sa jednocifrenim milisekundnim performansama na bilo kojoj skali. Kao baza podataka bez servera, plaća se samo ono što se iskoristi. DynamoDB skalira na nulu, nema hladnih pokretanja, nema nadogradnje verzije, nema prozora za održavanje, nema zastoja održavanja. Ova baza podataka nudi širok skup bezbednosnih kontrola i standarda usklađenosti. Za globalno distribuirane aplikacije, DynamoDB globalne tabele su multi-regionalna, multiaktivna baza podataka sa SLA dostupnošću od 99,999% i povećanom otpornošću. Pouzdanost DynamoDB-a je podržana upravljanim rezervnim kopijama i oporavkom u trenutku. Sa DynamoDB tokovima, moguća je izgradnja aplikacija vođenih događajima bez servera [4] [9].

3.1.4. Amazon SQS

Amazon Simple Queue servis (Amazon SQS) nudi siguran, izdržljiv i dostupan red podataka koji omogućava integraciju i odvojanje distribuiranih softverskih sistema i komponenti. Amazon SQS nudi uobičajene konstrukcije kao što su redovi za „mrtve“ poruke i oznake za alokaciju troškova. Pruža generički API za veb usluge kojima se može pristupiti koristeći bilo koji programski jezik koji podržava AWS SDK [10].

Amazon SQS razdvaja i skalira distribuirane softverske sisteme i komponente kao uslugu čekanja. Obično obrađuje poruke preko jednog pretplatnika, što je idealno za tokove posla gde su prevencija narudžbine i gubitka kritični. Za širu distribuciju, integracija Amazon SQS-a sa Amazon SNS-om omogućava razgranati obrazac za razmenu poruka, efektivno prosleđujući poruke većem broju pretplatnika odjednom [8].

3.1.5. Amazon Bedrock

Amazon Bedrock je servis koji omogućava da dostupnost osnovnih modela veštačke inteligencije visokih performansi (engl. Foundation Model) iz vodećih startapa veštačke inteligencije i Amazon za korisničku upotrebu putem objedinjenog API-ja. Moguće je birati između širokog spektra osnovnih modela veštačke inteligencije kako bi se pronašao model koji je najprikladniji za korisnikov slučaj upotrebe. Amazon Bedrock takođe nudi širok skup mogućnosti za izgradnju generativnih AI aplikacija sa bezbednošću, privatnošću i odgovornom veštačkom inteligencijom. Koristeći Amazon Bedrock, moguće je da se lako eksperimentiše nad i procenjuju osnovni modeli za korisničke slučajeve upotrebe, privatno da se prilagođavaju korisničkim podacima koristeći tehnike kao što su fino podešavanje i proširena generacija preuzimanja (RAG) i prave agenti koji izvršavaju zadatke koristeći sisteme korisnikovog preduzeća i izvore podataka.

Amazon Bedrock je rešenje bez servera, gde je moguće privatno prilagoditi osnovne modele sopstvenim podacima i lako i bezbedno ih integrisati i primeniti u aplikacije koristeći AWS alate bez potrebe za upravljanjem infrastrukturom i njenom izmenom [11].

4. PRINCIPI ARHITEKTURE SAMOOBNAVLJAJUĆEG KODA

AWS-ovi inženjeri su pioniri sistema za kod koji se sam obnavlja, pronalazi greške (engl. bag) i sam podiže zahteve za ispravku originalnog koda.

Arhitektura ovog rešenja pomaže softverskim kompanijama da postave sistem za otkrivanje i evidentiranje grešaka, generisanje ispravki grešaka i kreiranje zahteva za promenom koda. Svaka kompanija koja kreira softver neizbežno mora da uravnoteži rešavanje grešaka, a istovremeno se takmiči sa pritiskom razvoja proizvoda i funkcionalnosti. Greške mogu odvratiti pažnju programera, pogoršati korisničko iskustvo i uzrokovati pogrešne pokazatelje. Ovo idejno rešenje pomaže softverskim kompanijama da implementiraju automatizovani sistem koji otkriva i ispravlja greške kako bi poboljšao pouzdanost aplikacija i poboljšao celokupno korisničko iskustvo.

Rešenje za kod koji se sam popravlja je zasnovano na 6 osnovnih stubova najboljih arhitektonskih praksi za dizajniranje sistema u oblaku.

AWS Well-Architected Framework opisuje ključne koncepte, principe dizajna i najbolje arhitektonske prakse za projektovanje i pokretanje radnih opterećenja u oblaku. Ovi principi i koncepti su opisani detaljnije u radu [7].

8. ZAKLJUČAK

Inovativan pristup inženjera iz kompanije Amazon nam je donelo rešenje koje će, ukoliko se dobro podesi i iskoristi, dovesti do smanjivanja tehničkog duga i do olakšavanja i ubrzavanja toka razvoja aplikacije. Način na koji je iskorišćen generativni model veštačke inteligencije je

doveo do automatizacije velikog dela posla programera, a to je pronalaženje i ispravak grešaka u kodu.

Ovo rešenje smanjuje troškove razvoja softverskih rešenja, poboljšava produktivnost timova, obezbeđuje visok nivo sigurnosti i pouzdanosti,

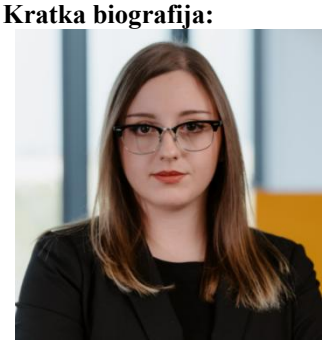
Potrebno je imati u vidu da je ovo rešenje dostupno samo u određenim AWS regionima i da se troškovi mogu značajno razlikovati u zavisnosti od obima upotrebe i konkretnih potreba korisnika. Pažljivo planiranje i praćenje troškova, uz korišćenje AWS alata kao što je Cost Explorer, omogućava korisnicima da maksimiziraju koristi od ovog sistema uz minimalne finansijske rizike

Na kraju, ovakve vrste rešenja predstavljaju budućnost razvoja softvera, gde automatizacija i veštačka inteligencija igraju ključnu ulogu u poboljšanju kvaliteta softverskih rešenja i korisničkog iskustva. Ovakve tehnologije su već krenule da transformišu način na koji timovi softverskih inženjera rade, smanjujući tehnički dug i omogućavajući brže i efikasnije rešavanje problema.

9. LITERATURA

[1] Funkcija kao servis [na mreži]. Dostupno na: https://www.ibm.com/topics/faas (Pristupljeno u junu 2024. godine)
[2] Lambda Servis [na mreži]. Dostupno na: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html (Pristupljeno u junu 2024. godine)
[3] Proizvodi AWS-a [na mreži]. Dostupno na: https://aws.amazon.com/products/ (Pristupljeno u junu 2024. godine)
[4] AWS načini skladištenja podataka [na mreži]. Dostupno na: https://www.educba.com/aws-storage-services/ (Pristupljeno u junu 2024. godine)
[5] O bazama podatala [na mreži]. Dostupno na: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html (Pristupljeno u junu 2024. godine)
[6] Self healing code uputstvo [na mreži]. Dostupno na: https://aws.amazon.com/solutions/guidance/self-healing-code-on-aws/ (Pristupljeno u junu 2024. godine)
[7] Principi arhitekture na AWS-u [na mreži]. Dostupno na: https://aws.amazon.com/architecture/well-architected/?wa-lens-whitepapers.sort-by=item.additionalFields.sortDate&wa-lens-whitepapers.sort-order=desc&wa-guidance-whitepapers.sort-by=item.additionalFields.sortDate&wa-guidance-whitepapers.sort-order=desc (Pristupljeno u junu 2024. godine)
[8] Cloudwatch [na mreži]. Dostupno na: https://aws.amazon.com/cloudwatch/ (Pristupljeno u junu 2024. godine)
[9] DynamoDB [na mreži]. Dostupno na: https://aws.amazon.com/dynamodb/ (Pristupljeno u junu 2024. godine)
[10] Simple Queue Service [na mreži]. Dostupno na: https://docs.aws.amazon.com/AWSSimpleQueueService/ (Pristupljeno u junu 2024. godine)
[11] Bedrock [na mreži]. Dostupno na: https://docs.aws.amazon.com/bedrock/ (Pristupljeno u junu 2024. godine)
[12] Self healing code [na mreži]. Dostupno na: https://aws-solutions-library-samples.github.io/ai-ml/self-healing-code-on-aws.html (Pristupljeno u junu 2024. godine)

Kratka biografija:



Dragana Trivunović rođena je 21. oktobra 1998. godine u Sremskoj Mitrovici. Osnovnu školu "Čaki Lajoš" završila je u Bačkoj Topoli, a gimnaziju, opšti smer, je završila u srednjoj školi "Dositej Obradović" u Bačkoj Topoli. Školske 2017/2018 godine upisuje Fakultet tehničkih nauka u Novom Sadu, na studijski program Primenjeno softversko inženjerstvo. Osnovne akademske studije završila je školske 2022/2023, posle kojih upisuje master studije, takođe, na studijskom programu Primenjeno softversko inženjerstvo.