Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/29BE04Kolovic

PROJEKTOVANJE INFRASTRUKTURE ZA POTREBE KLIJENT-SERVER APLIKACIJE I EVALUACIJA PROJEKTOVANE INFRASTRUKTURE KROZ PRIZMU DOBROG ARHITEKTONSKOG OKVIRA

INFRASTRUCTURE DESIGN FOR CLIENT-SERVER APPLICATIONS AND EVALUATION OF DESIGNED INFRASTRUCTURE THROUGH THE PRISM OF A WELL ARCHITECTURAL FRAMEWORK

Nikola Kolović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U sklopu ovog rada je projektovana i implementirana infrastruktura za potrebe klijent-server aplikacije na AWS platformi. Opisani su korišćeni servisi na kojima se infrastruktura zasnika, i prikazani su detalji implementacije servisa. Dodatno, izvršena je analiza i evaluacija projektovane infrastrukture kroz prizmu dobrog arhitektonskog okvira koji AWS platforma obezbeđuje prema dva osnovna stuba: bezbednost i operativna efikasnost. Dobijeni rezultat rada predstavlja sigurnu i skalabilnu infrastrukturu za potrebe klijent server aplikacija, koja je lako proširiva i jednostavna za nadogradnju koje su predložene rezulatima evaluacije.*

Ključne reči: *Amazon Web Services, Računarstvo u oblaku, Terraform*

Abstract – *As part of this work, the infrastructure for the client-server application on the AWS platform was designed and implemented. The services used on which the infrastructure is based are described, and the details of the service implementation are presented. Additionally, an analysis and evaluation of the designed infrastructure was performed through the prism of a well architected framework provided by the AWS platform according to two pillars: security and operational efficiency. The resulting work represents a secure and scalable infrastructure for the needs of client-server applications, which is easily expandable and simple to upgrade, as suggested by the evaluation results.*

Keywords: *Amazon Web Services, Cloud computing, Terraform*

1. UVOD

U savremenom svetu, svedoci smo ubrzanog rasta informacionih tehnologija, i sve veće potrebe za kompleksnijim aplikacijama koje treba da opsluže milione korisnika. Tradicionalne aplikacije, koje su se bazirale na monolitnom pristupu, i lokalnim infrastrukturama postaju irelevantne u odnosu na brzo-rastuća rešenja u oblaku.

Računarstvo u oblaku donosi novi pristup kreiranja infrastrukture koji omogućavaju jednostavnije kreiranje i održavanje infrastrukture, a krajnjim korisnicima bolje korisničko iskustvo. Pored jednostavnijeg kreiranje infrastrukture, postoje mnogi benefiti računarstva u oblaku kao što su povećana bezbednost, mogućnost skaliranja resursa, visoka dostupnost servisa, robusnost sistema i druge.

Ovaj rad baviće se projektovanjem infrastrukture u oblaku na AWS (Amazon Web Services) [1] platformi, i implementacija korišćenjem IaaC (Infrastructure as a Code) prakse za kreiranje resursa u oblaku. Implementirana infrastruktura će biti namenjena klijent-server aplikacijama. Analiziraćemo projektovanu infrastrukturu kroz prizmu AWS Well-Architected Framework-a [2], gde ćemo se fokusirati na dva stuba ovog okvira: bezbednost i operativnu efikasnost. Čilj je da se projektuje i implementira infrastuktura koja je dovoljno fleksibilna za buduće izazove, a istovremeno bezbedna, skalabilna i pouzdana.

U drugom poglavlju predstavljeni su AWS servisi i tehnologije koje su korišćene u radu. Opis i detalji implementacije projektovane infrastrukture prikazani su u trećem poglavlju. Četvrto poglavlje sadrži evaluaciju projektovane infrastrukture kroz prizmu dobrog arhitektonskog okvira. U petom poglavlju je iznet zaključak gde su opisane prednosti i mane projektovane infrastrukture, i izneti predlozi za dalji razvoj.

2. KORIŠĆENE TEHNOLOGIJE I AWS SERVISI

Za implementaciju infrastrukture koristili smo alat Terraform [3] koji je trenutno industrijski standard za obezbeđivanje infrastrukture pisanjem programskog koda.

AWS servisi koje smo koristili i koji integrisani zajedno zadovoljavaju potrebe velićine klijent-server aplikacija:

1. AWS Identity and Access Management - koristi se za kreiranje identiteta i upravljanjem dozvolama na platformi
2. AWS Organization - omogućava centralnu organizaciju naloga u okruženjima sa više naloga
3. AWS Identity Center - omogućava centralno upravljanje korisnicima i dozvolama u okruženjima sa više naloga

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio docent Željko Vuković, red. prof.

4. AWS Virtual Private Cloud - koristi se za kreiranje virtuelne privatne mreže radi izolacije resursa, kao i za konfiguraciju dodatnih sigurnosnih postavki
5. AWS Elastic Compute Cloud - omogućava kreiranje virtuelnih mašina u oblaku
6. AWS Elastic Load Balancer - koristi se za kreiranje i konfiguraciju balansera opterećenja
7. AWS Elastic Container Service - omogućava pokretanje i orkestraciju kontejnerizovanih aplikacija u oblaku
8. AWS Relational Database Service - koristi se za kreiranje i upravljanje relacionim bazama podataka
9. AWS Simple Storage Service - predstavlja visokodostupno, skalabilno skladište podataka
10. AWS Cloudfront - omogućava prenos statičkog i dinamičkog saobraćaja ultra brzim mrežama za prenos sadržaja
11. AWS Elastic Container Registry - omogućava čuvanje i upravljanje Docker slikama
12. AWS Secret Manager - koristi se za čuvanje, kao i za rotiranje tajni

3. OPIS PROJEKTOVANE INFRASTRUKTURE

Dizajn rešenja koje je implementirano na AWS platformi je bio poseban izazov zbog potrebe za visokim nivoom skalabilnosti, dostupnosti i sigurnosti. Cilj je bio da se kreira rešenje dovoljno fleksibilno da podrži rastuće zahteve aplikacije, a istovremeno dovoljno robusno da prihvati različite vrste opterećenja.

3.1 Dizajn infrastrukture

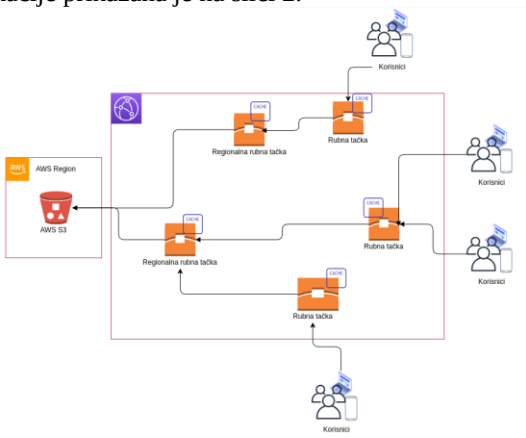
Dijagram na slici 1. [4, 5] prikazuje servise koji su korišćeni i njihov raspored unutar virtuelne privatne mreže, čime smo želeli da obezbedimo sigurno okruženje sa sto manje manjkavosti i sigurnosnih propusta. Za implementaciju virtuelne privatne mreže koristili smo AWS VPC, kojim smo kreirati tri grupe podmreža: javne, privatne, i izolovane. U podmreže su smešteni resursi zavisno od potrebe za internet konekcijom. Važno je napomenuti da je svaki tip podmreže kreiran u dve zone dostupnosti, čime obezbeđujemo visoku dostupnost aplikacije.

Bezbednosne grupe su postavljene tako da bazi podataka mogu pristupiti samo servisi iz privatne podmreže, i dodatno adresa bastion host-a čime obezbeđujemo pristup bazi podataka razvojnom timu. Servisima u privatnoj podmreži mogu pristupiti servisi iz javne podmreže, dok smo za servise u javnoj podmreži prilagodili bezbednosne grupe potrebama. Tako balanseru opeterećenja možemo pristupiti samo na portovima 80 i 443, dok bastion host-u možemo pristupiti samo na portu 22.

Za potrebe baze podataka, koristili smo AWS RDS servis, gde smo kreirali PostgreSQL bazu podataka, koja je smeštena u izolovanoj podmreži. Za potrebe upravljanja kredencijalima baze podataka koristili smo AWS Secret Manager, koji čuva i rotira kredencijale za pristup.

Serverski deo aplikacije se pokreću u kontejnerizovanom okruženju upravljanom AWS ECS servisom, koji se nalaze u privatnoj podmreži. Konektuju se na bazu podataka korišćenjem kredencijala iz tajni, dok se ispred servisa nalazi balanser opterećenja zadužen za raspoređivanje dolaznog saobraćaja.

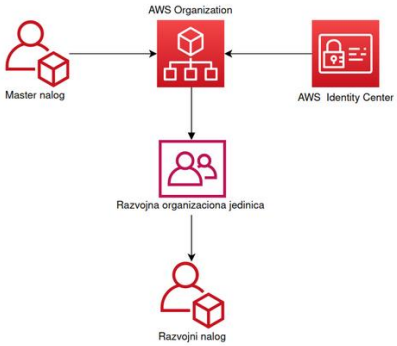
Klijentski deo aplikacije predstavlja statički veb sajt, čiji se fajlovi nalaze u S3 kanti, dok se isporučuju krajnjim korisnicima korišćenjem AWS Cloudfront distribucije, koja omogućava keširanje i brzu isporuku saobraćaja kroz mrežu rubnih tačaka. Ovakva konfiguracija klijentske aplikacije prikazana je na slici 2.



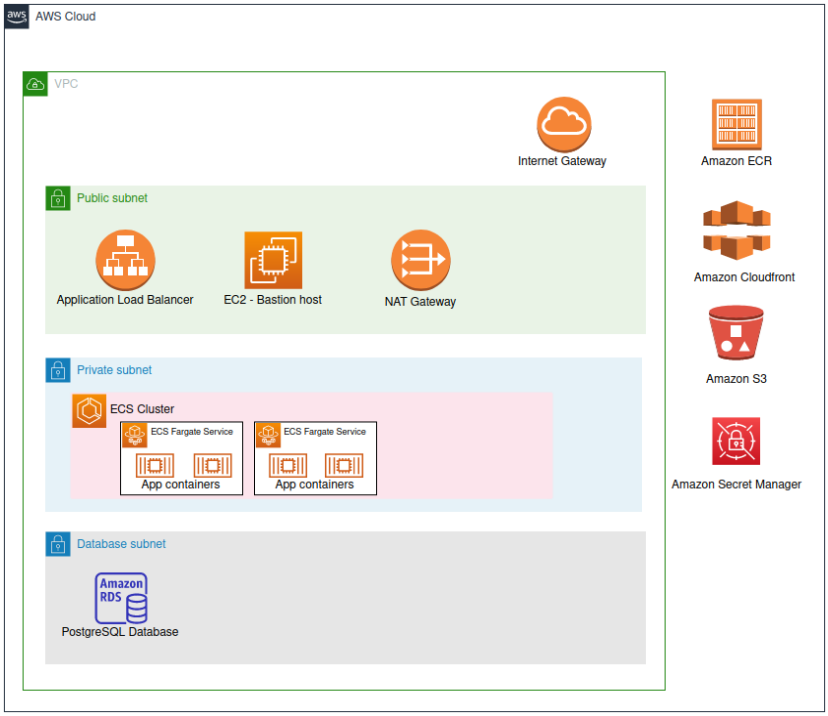
Slika 2. *Prikaz dizajna rešenja za isporuku klijentske aplikacije*

Da bismo obezbedili fleksibilnost i bezbednost sistema, koristili smo okruženje sa više naloga. U trenutnom obimu rada potrebni su nam samo razvojni i master nalog, ali je dizajn dovoljno fleksibilan da u buducnosti može podržati dodatna okruženja na različitim nalozima jednostavnim izmenama u Terraform konfiguracionim fajlovima.

Dijagram na slici 3. prikazuje rešenje implementirano sa dva AWS naloga: master i nalog za razvojno okruženje. Master nalog se koristi za upravljanje okruženjima sa više naloga pomoću servisa AWS Organization i AWS Identity Center. Za potrebe razvojnog okruženja kreiran je dodatni nalog koji je smesten u organizacionoj jedinici za razvojne naloge.



Slika 3. *Prikaz dizajna okruženja sa više naloga*



Slika 1. *Prikaz dizajnirane infrasturkture za potrebe jednog okruženja*

3.2 Konfiguracija Terraform okruženja

Kada je u pitanju organizacija konfiguracionih fajlova Terraform okruženja, na najvišem nivou koristili smo dva direktorijuma:

1. project-infrastructure - sadrži direktorijume za pozive modula, gde je opisano željeno stanje na AWS platformi
2. tf-modules - sadrži direktorijume sa kreiranim ponovno iskoristivim modulima

U project-infrastructure direktorijumu, podelili smo servise u zasebne direktorijume, gde smo definisali Terraform provajder koji koristimo, kao i konfiguraciju za čuvanje i enkripciju Terraform stanja.

Za potrebe čuvanja Terraform stanja, koristili smo udaljeni backend koji se nalazi na S3 kanti. Dodatno, omogućili smo zaključavanje stanja, i enkripciju datoteke za čuvanje stanja.

4. EVALUACIJA PROJEKTOVANE INFRASTRUKTURE KROZ PRIZMU DOBROG ARHITEKTONSKOG OKVIRA

Dobar arhitektonski okvir, ili AWS Well Architected Framework, predstavlja skup principa i najboljih praksi kreiranih od strane AWS, kako bi pomogao korisnicima da implementiraju sigurne, visoko dostupne, efikasne i robusne aplikacije u oblaku. Okvir se sastoji od pet stubova, koji uključuju principe koje treba slediti i smernice za njihovu implementaciju, kako bi osigurali da infrastruktura u oblaku bude izgrađena na najbolji mogući način.

Okvir se sastoji od 5 stubova:

1. Bezbednost (Security)
2. Operativna efikasnost (Operational Excellence)
3. Pouzdanost (Reliability)
4. Performanse (Performance Efficiency)
5. Optimizacija troškova (Cost Optimization)

Analizom svakog od ovih stubova, mogu se identifikovati potencijalni nedostaci i mogućnosti za unapređenje infrastrukture.

Fokus ovog rada su bila prva dva okvira: bezbednost i operativna efikasnost.

4.1 Bezbednost

Bezbednost predstavlja jedan od najbitnijih stubova i predstavlja temelj svake uspešne infrastrukture u oblaku. Bezbednost u oblaku obuhvata zaštitu celokupnog radnog okruženja, uključujući podatke, sisteme i sve resurse.

Zasniva se na 7 osnovnih principa:

1. Implementacija čvrste osnove identiteta
2. Održavanje praćenja
3. Primena bezbednosti na svim slojevima
4. Automatizacija najboljih bezbednosnih praksi
5. Zaštita podataka u tranzitu i u mirovanju
6. Održavanje ljudi daleko od podataka
7. Priprema za sigurnosne događaje

Evaluacijom projektovane infrastrukture u kontekstu stuba bezbednosti zaključili smo da projektovana infrastruktura u najvećoj meri zadovoljava preporuke navedenih principa.

Pre svega, osnova identiteta je implementirana praćenjem najboljih praksi. Odnosno, kreiranjem okruženja sa više naloga, sa centralnim upravljanjem identitetima i dozvolama, što smo postigli pomoću AWS Organization i AWS Identity Center servisa. Dodatno, dozvole smo u određenoj meri kreirali poštovanjem principa najmanjih privilegija i obezbedili smo privremene kredencijale za pristup.

Bezbednost na nivou okruženja je takođe uglavnom implementirana u skladu sa smernicama definisanim od strane AWS. Osnovu mrežne bezbednosti smo implementirali pomoću AWS VPC servisa, gde smo definisali bezbednosne grupe na nivou servisa.

Bezbednost podataka smo obezbedili enkripcijom podataka u bazi podataka, na S3 kantama kao i tajnama. Dodatno, razvojni tim nema direktan pristup bazi podataka i imamo implementirane sigurnosne kopije podataka.

Kompletna infrastruktura je implementirana u skladu sa praksom pisanja infrastructure kroz kod, koristeći Terraform alat.

Preporuke za unapređenje će biti iznete u zaključku ovog rada.

4.2 Operativna efikasnost

Operativna efikasnost predstavlja značajan stub za kreiranje uspešne infrastrukture. Operativna efikasnost uključuje optimizaciju operacija, kontinualno unapređivanje procedura, i održavanje visokog nivoa performansi i pouzdanosti.

Zasniva se na 5 osnovnih principa:

1. Izvršavanje operacija kroz kod
2. Pravljenje čestih, malih, revizibilnih promena
3. Često unapređivanje operativnih procedura
4. Korišćenje upravljivih servisa
5. Implementacija nadzora radi sticanja korisnih uvida

Za razliku od stuba bezbednosti, gde je infrastruktura u najvećoj meri zadovoljavala principe, u ovom slučaju su principi delimično zadovoljeni, pre svega iz razloga što u trenutnoj fazi projekta, neki principi ne mogu biti implementirani, dok za neke druge nemamo nikakvih prednosti za samu infrastrukturu.

Uprkos tome, veliki deo principa je zadovoljen. Infrastruktura je implementirana kroz kod, prilagođavanjem ponovno iskoristivih modula, što dodatno implicira da smo omogućili kreiranje malih promena na nivou servisa, bez uticaja na druge servise. Korišćenjem alata za kontrolu verzija Git, omogućeno je verzionisanje i praćenje promena.

Uglavnom su korišćeni upravljivi servisi, čime smo omogućili razvojnim timovima da budu efikasniji i da se oslobode dodatnih zadataka za održavanje infrastrukture.

Preporuke za unapređenje će biti iznete u zaključku ovog rada.

5. ZAKLJUČAK

Na osnovu projektovane infrastrukture koja predstavlja složeno ali istovremeno i dovoljno fleksibilno rešenje, dobijena je funkcionalna infrastruktura, koja je apsolutno operativna, bezbedna, i dovoljno skalabilna za opsluživanje velikog broja korisnika.

Evaluacijom implementirane infrastrukture kroz prizmu dobrog arhitektonskog okvira zaključili smo da infrastruktura ima jako dobro dizajniranu osnovu, korišćenjem AWS okruženja sa više naloga, i centralnim servisom za upravljanje identitema i dozvolama, što predstavlja najbolje prakse u vidu fundamentalne bezbednosti i buduće proširivosti. Bezbednost podataka i drugih komponenti je na visokom nivou, sledili smo najbolje prakse za konfigurisanje privatne mreže, i zaštitili sve slojeve. Dodatno, enkriptujemo podatke u tranzitu i u mirovanju gde je to bilo moguće, automatski rotiramo tajne, i pravimo redovne rezervne kopije podataka. Instrastuktura je obezbeđenja pisanjem programskog koda, čime smo osigurali da je infrastruktura uvek u željenom stanju, i dodatno standardizovali kako operativne tako i sigurnosne postavke kreiranjem Terraform modula.

S obzirom da je infrastruktura implementirana korišćenjem najmodernijih tehnologija, i servisa koji se konstantno razvijaju i unapređuju, sasvim je legitimno reći da implementirano rešenje ima prostora za dalja usavršavanja, što je pokazala i evaluacija projektovane infrastrukture.

Jedan od segmenata za unapređenje je svakako zakupljivanje privatnog domena, čime bismo rešili problem sa kreiranjem SSL sertifikata na nivou balansera opterećenja. Rešenje za pristup bazi podataka može biti poboljšano uključivanjem AWS Session Manager-a i aktiviranjem funkcionalnosti za IAM autentikaciju na nivou AWS RDS baze podataka, koja obezbeđuje privremene kredencijale za korisnike koji imaju dozvole definisane politikama.

Dodatno se može obratiti paznja na reviziciju principa najmanjih privilegija, i sistem praćenja metrika i upozorenja.

6. LITERATURA

[1] AWS - Amazon Web Services - https://aws.amazon.com/ (pristupljeno u aprilu 2024.)

[2] AWS Well-Architected Framework - https://aws.amazon.com/architecture/well-architected/ (pristupljeno u aprilu 2024.)

[3] Terraform - https://www.terraform.io/use-cases/infrastructure-as-code/ (pristupljeno u aprilu 2024.)

[4] James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach”, 2017.

[5] Gauvar Agarwal, “Modern DevOps Practices”, 2021.

Kratka biografija:

Nikola Kolović rođen je u Kraljevu 1998. god. Osnovnu i srednju školu završio je u Kraljevu, nakon čega je 2017. godine upisao Fakultet tehničkih nauka u Novom Sadu. Diplomirao je 2021. na modulu primenjene računarske nauke i iste godine upisao Master studije.
Kontakt: nikola.kolovic7@gmail.com