Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.738.5:339
DOI: https://doi.org/10.24867/31BE32Fruza

# PROJEKTOVANJE OSNOVNE INFRASTRUKTURE ZA HOSTOVANJE VEB APLIKACIJA

# DESIGNING OF THE BASIC INFRASTRUCTURE FOR HOSTING WEB APPLICATIONS

Denis Fruža, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj –** U ovom radu je prikazan proces dizajniranja i implementacije osnovne *AWS cloud* infrastrukture za hostovanje veb aplikacija uz korišćenje *Terraform IaC (Infrastructure as code) alata. Akcenat je stavljen na postavljanje visoko dostupne, skalab*ilne i bezbedne infrastrukture. Takođe su opisane i najbolje prakse za projektovanje *VPC*-a, upravljanje resursima kao i upotreba nekih od bitnih *AWS* servisa.

**Ključne reči:** *cloud tehnologije, infrastruktura, internet mreže, AWS, Terraform, veb aplikacija*

**Abstract** – *This paper presents the process of designing and implementing the basic AWS cloud infrastructure for hosting web applications using Terraform IaC (Infrastructure as Code) tools. The focus is on setting up a highly available, scalable, and secure infrastructure. Best practices for VPC design, resource management, as well as the use of some essential AWS services, are also described.*

**Keywords:** *cloud technologijes, infrastructure, internet networks, AWS, Terraform, web application*

## 1. UVOD

U trenutnoj fazi digitalne revolucije, *cloud computing* se istakao kao nezaobilazna komponenta svake *IT* infrastrukture. On omogućava organizacijama da lako upravljaju svojim resursima, optimizuju troškove i unaprede dostupnost svojih aplikacija. *Amazon Web Services* (*AWS*) se nalazi na čelu ove industrije i svojim korisnicima nudi preko 200 različitih servisa iz globalno raspoređenih *data* centara.

Iako *cloud* servisi pojednostavljuju procese, upravljanje složenijim sistemima može biti veoma izazovno što naglašava značaj upravljanja infrastrukturom uz pomoć *infrastructure as code* (*IaC*) pristupa. Terraform, kao vodeći alat u ovoj oblasti, omogućava automatizaciju i efikasno upravljanje *cloud* infrastrukturom kroz kod.

Na početku rada biće prikazane osnove cloud computinga, ključni *AWS* servisi i najbolje prakse za njihovu upotrebu, s posebnim naglaskom na *Virtual Private Cloud* (*VPC)* i mrežnu bezbednost kao osnovu za kreiranje visoko dostupne, skalabilne i bezbedne infrastrukture. Takođe će biti objašnjen i *IaC* koncept i korišćenje Terraform-a za automatizaciju i upravljanje infrastrukturom. Nakon potrebnog predznanja, detaljno će biti prikazan proces dizajniranja i implementacije infrastrukture za razvoj i skaliranje veb aplikacija.

## 2. TEORIJSKE OSNOVE

*Cloud computing* predstavlja skup tehnologija koje se koriste za isporuku računarskih resursa kojima mogu pristupiti korisnici širom sveta. Osnovna ideja je omogućiti pristup podacima bez obzira na trenutno okruženje korisnika. Krajnji korisnici pristupaju *cloud* aplikacijama putem veb, mobilnih ili desktop aplikacija. Postoje tri tipa servisa u *cloud computing*-u:

- infrastruktura kao servis (IaaS) - *cloud* provajder upravlja fizičkom infrastrukturom, a klijent ima mogućnost da na toj infrastrukturi upravlja operativnim sistemom, mrežom i skladištem. Nudi najširi spektar mogućnosti i pristupa.
- platforma kao servis (PaaS) - *cloud* provajder klijentima dodatno pruža i radni okvir (*framework*), aplikativne programske interfejse (*API*) i druge alate neophodne za razvoj, isporuku i testiranje aplikacija. Klijent upravlja svojim aplikacijama koje instalira na datoj infrastrukturi i u određenoj meri njihovom konfiguracijom.
- softver kao servis (SaaS) - klijentima (korisnicima aplikacija) pruža se gotov softver na korišćenje. Provajder tog softvera u potpunosti upravlja infrastrukturom i aplikacijama na infrastrukturi.

Neki od popularnih *cloud* servis provajdera su *Amazon Web Services* (*AWS*), *Microsoft Azure*, *Google Cloud Platform* (*GCP*) i *Digital Ocean*.

---

**Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Željko Vuković**

### 2.1. *AMAZON WEB SERVICES* (*AWS*)

*Amazon Web Services* (*AWS*) predstavlja *cloud* platformu veb servisa koja pruža kompletna rešenja u mnogim domenima računarstva kao što su skladištenje i održavanje baza podataka, obrada samih podataka, pružanje infrastrukture za najkompleksnija rešenja, mašinsko učenje i drugo. Ukupan broj servisa koje pruža je preko 200 i u 2022. godini zauzimao je 34% čitavog *cloud* tržišta.

### 2.2. INFRASTRUKTURA KAO KOD

Infrastruktura kao kod (*IaC*) predstavlja moderan pristup pri upravljanju infrastrukturom. Ovaj koncept transformiše manuelni u automatizovani pristup upravljanja infrastrukturom uz pomoć koda. Kompletna konfiguracija i postavka infrastrukture opisana je u lako čitljivim konfiguracionim fajlovima. *IaC* u velikoj meri pojednostavljuje procese implementacije, održavanja i skaliranja infrastrukture. Najpopularniji *IaC* alati su *Terraform*, *Puppet*, *AWS CloudFormation*, Ansible i drugi.

## 3. *AWS* I NJEGOVI SERVISI

Neki od najpopularnijih servisa na *AWS*-u kao što su *Virtual Private Cloud* (*VPC*), Elastic *Compute Cloud* (*EC2*), *Relational Database Service* (*RDS*), *Elastic Load Balancing* i *Autoscaling, Simple Storage Service* (*S3*) kao i drugi servisi korišćeni su u projektovanju i implementaciji infrastrukture u ovom master radu. U daljem tekstu biće opisani neki od korišćenih servisa:

- *Virtual Private Cloud* (*VPC*) - predstavlja logički izolovanu privatnu mrežu u kojoj se postavljaju resursi. Dozvoljava kreiranje podmreža, konfigurisanje IP adresa, bezbednosnih zaštita, rutiranja saobraćaja i omogućava potpunu kontrolu nad mrežom.
- *Elastic Compute Cloud* (*EC2*) - servis koji omogućava zauzimanje virtuelnih mašina (*EC2* instanci) za potrebe hostovanja aplikacija, skladištenja, obrade i analize podataka itd.
- *Relational Database Service* (*RDS*) - servis za upravljanje relacionim bazama podataka. Omogućava veoma jednostavno postavljanje, upravljanje i skaliranje relacionih baza podataka kao što su *MySQL*, *PostgreSQL*, *Amazon Aurora*, *MariaDB* itd. Sve bitne operacije poput pravljenja rezervnih kopija (*backup*-a), replikacija, oporavaka od grešaka ili vraćanja baze podataka obavljaju se automatski od strane *AWS*-a.
- *Elastic Load Balancing* i *Autoscaling* – predstavljaju servise koji imaju ulogu u postizanju maksimalne dostupnosti i skalabilnosti infrastrukture. *Load Balancer*-i vrše distribuciju saobraćaja na više različitih ciljeva unutar jedne ili više zona dostupnosti. *Auto Scaling* omogućava automatsko prilagođavanje broja resursa (npr. *EC2* instanci) na osnovu unapred definisanih uslova i metrika i ima veoma važnu ulogu kako u skaliranju infrastrukture, tako i u optimizaciji troškova.
- *Simple Storage Service* (*S3*) - predstavlja skalabilan, visoko dostupan i siguran servis za skladištenje podataka. Namenjen je za skladištenje i dobavljanje velike količine podataka po veoma niskoj ceni. Nema nikakve limite kada su u pitanju količina skladištenih podataka kao i broj konkurentnih preuzimanja istih.

## 4. *TERRAFORM* - INFRASTRUKTURA KAO KOD

*Terraform* je alat otvorenog koda koji se koristi za upravljanje infrastrukturom. Omogućava definisanje i upravljanje infrastrukturom putem koda (*IaC*). Kod predstavljaju konfiguracioni fajlovi napisani u *HashiCorp Configuration Language*-u (*HCL). Terraform* u velikoj meri olakšava i ubrzava upravljanje infrastrukturom za razliku od manuelnog upravljanja. Sve *terraform* datoteke imaju *.tf* ekstenziju i sadrže definicije resursa, njihove karakteristike, odnose sa drugim resursima kao i druge parametre koji opisuju stanje infrastrukture.

*HashiCorp Configuration Language* (*HCL*) predstavlja deklarativni jezik koji je posebno dizajniran za *Terraform.* Dizajniran je tako da bude veoma jednostavan za čitanje i pisanje i strukturiran je tako da podseća na *JSON* format za razmenu podataka.

### 4.1. SINTAKSA I OSNOVNE KOMANDE

Osnovni deo sintakse za definisanje i konfigurisanje resura su Terraform blokovi. Blokovi pružaju strukturiran način organizovanja i specificiranja komponenti. Neki od najvažnijih terraform blokova su *terraform, provider, resource, data, locals i module.*

Osnovne komande koje se u *Terraform*-u koriste za interakciju sa infrastrukturnim resursima su:

- *terraform init* – koristi se za inicijalizaciju radnog direktorijuma i povlačenje svih potrebnih *plugin*-ova i provajdera definisanih u konfiguraciji.
- *terraform plan* – kreira pregled inkrementalnih promena na osnovu promenjene konfiguracije. Analizira trenutno stanje infrastrukture i poredi ga sa željenim stanjem.
- *terraform apply* - koristi se za primenu promena opisanih u izvršnom planu *Terraform*-a. Vrši kreiranje, promenu ili brisanje postojećih resursa.
- *terraform destroy* - vrši uništavanje svih resursa definisanih u *Terraform* konfiguraciji. Ova komanda je nepovratna pa je treba oprezno koristiti.

### 4.2. *TERRAFORM STATE FILE*

*Terraform state file* predstavlja fajl u kojem se čuva trenutno stanje infrastrukture. Ovaj fajl omogućava praćenje metapodataka i olakšava radnje poput primene

novog koda, uništavanje postojeće infrastrukture ili postavljanja novih resursa. Može se čuvati lokalno, ali se preporučuje udaljeno skladištenje na takozvanom t*eraform state backend*-u. Na *AWS*-u kao *state backend* koristi se *AWS S3 bucket*.

## 5. BEZBEDNOST NA *AWS*-U

Prva linija odbrane kada je u pitanju bezbednost na *cloud*-u predstavlja filtriranje ulaznog i izlaznog saobraćaja *VPC*-a. *AWS* nudi nekoliko servisa za filtriranje ulaznog i izlaznog saobraćaja kao što su sigurnosne grupe, *Network Access Control List i VPC Flow Logs* za praćenje logova:

- sigurnosne grupe – virtuelni *firewall* koji kontroliše sav ulazni i izlazni saobraćaj. Saobraćaj se kontroliše tako što se definišu ulazna i izlazna pravila. Sigurnosna grupa se može povezati sa različitim resursima kao što su *EC2*, *RDS* ili *AWS Load Balancer*
- network access control list – dodatni sloj sigurnosti koji predstavlja virtuelni *firewall* za kontrolisanje ulaznog i izlaznog saobraćaja podmreža u *VPC*-u. Pravila koja se definišu veoma su slična onim kod sigurnosnih grupa osim što nemaju stanje (*stateless*) i zahtevaju posebno kreiranje ulaznog i izlaznog pravila kako bismo dozvolili određenu komunikaciju.
- *VPC* flow logs - omogućava praćenje toka saobraćaja unutar *VPC*-a. Može se kreirati za ceo VPC, podmrežu ili mrežni interfejs.

### 5.1. *WEB APPLICATION FIREWALL (WAF)*

*Web Application Firewall* (*WAF*) predstavlja servis koji ima ulogu da zaštiti vec aplikacije od uobičajenih napada i botova. Pomaže u kreiranju listi za kontrolu pristupa i raznovrsnih pravila u vidu uslova koja se primenjuju protiv sigurnosnih pretnji. Neki od najčešćih napada od kojih nudi zaštitu uključuju *SQL Injection, Cross-Site Scripting (XSS), DDoS,* razni skeneri i botovi. Osim ovih definisanih pravila, *AWS* nudi i setove upravljanih pravila koja možemo naći na *AWS Marketplace*-u. Jedan od najvažnijih i najpopularnijih setova pravila je *OWASP Top 10* koji predstavlja listu deset najčešćih sigurnosnih rizika na veb aplikacijama.

## 6. PROJEKTOVANJE I IMPLEMENTACIJA INFRASTRUKTURE

Projektovanje i implementaciju infrastrukture potrebno je započeti detaljnom analizom same aplikacije i njenih funkcionalnosti kako bi se na osnovu toga postavila odgovarajuća arhitektura. Osim toga, razmatranje budžeta, veličine tima koji radi na razvoju kao i analiza geografske lokacije krajnjih korisnika aplikacije uticaće na odabir *AWS* regiona kao i na inicijalan broj okruženja. U ovom slučaju to će biti us-east-1 region i tri okruženja: *development, staging i production.*

### 6.1. PRIPREMA ZA IMPLEMENTACIJU I NAJBOLJE PRAKSE

Implementacija je započeta konfigurisanjem *Terraform*-a. Za svako okruženje kreiran je S3 bucket koji će se koristiti kao *terraform remote backend* kao i *AWS DynamoDB* tabela koja se koristi za zaključavanje fajla u kojem se čuva trenutno stanje infrastrukture kako bi se izbegli konflikti. Za verzionisanje *Terraform* koda i efikasno upravljanje istim, koristiće se *Github*.

Prilikom projektovanja *VPC*-a, osnovne komponente infrastrukture, od ključnog je značaja pridržavati se najboljih praksi u samom startu kako bi se infrastruktura bez poteškoća skalirala u budućnosti. Takođe, određene konfiguracije na *VPC*-u nije moguće menjati nakon kreiranja. Neke od najboljih praksi predstavljaju:

- korišćenje najvećeg *CIDR* bloka i jedinstvenog opsega *IP* adresa
- korišćenje *Elastic IP* adresa
- korišćenje tagova
- korišćenje *VPC Peering*-a
- planiranje skalabilne infrastrukture u samom početku implementacije

### 6.2. ANALIZA KREIRANIH RESURSA

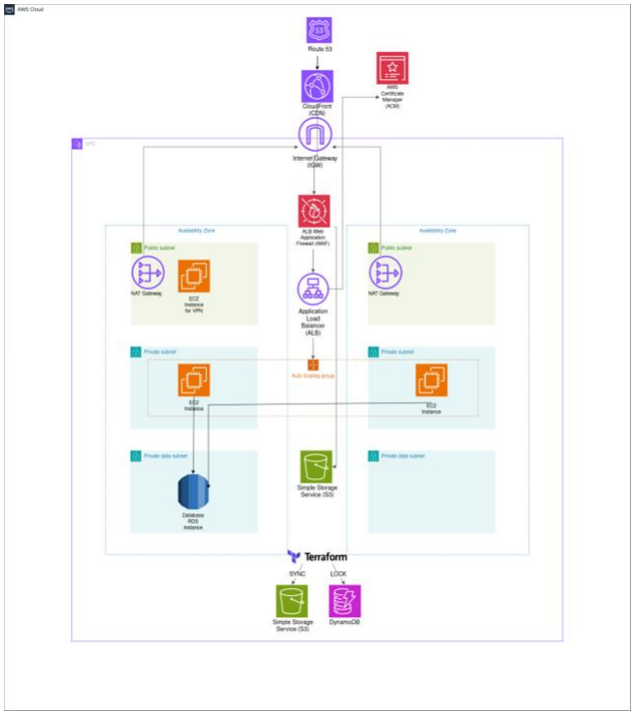
Za odabrani region najpre je kreiran *VPC*. Za produkciono okruženje odabran je 10.10.0.0/16 *CIDR* blok koji omogućava kreiranje $2^{16}$ odnosno 65536 hostova. *VPC* je podeljen na 6 podmreža, po jedna *public, private i private data*, u dve različite zone dostupnosti. Zatim su kreirane i ostale mrežne komponente kao što su tabele rutiranja, *NAT* i *Internet Gateway* za preslikavanje IP adresa i pristup internetu.

Sigurnosne grupe kreirane su uz poštovanje strogih pravila (*least privilege*, specifični opsezi *IP* adresa) koja nalažu da se svim resursima omogući samo neophodan saobraćaj. Dodatno je kreiran i *Wireguard VPN* server kako bi se *SSH* pristup serverima kao i pristup bazama podataka ograničio na privatnu mrežu. Osim sigurnosnih grupa, na infrastrukturi je radi bezbednosti implementiran i *WAF (Web Application Firewall*) koji štiti od najčešćih napada i sigurnosnih propusta.

Statički *front-end* veb sajt nalaziće se na *S3 bucket*-u koji je kreiran za njegovo hostovanje. Za distribuiranje statičkog sadržaja kreirana je *CloudFront* (*CDN*) distribucija.

Kako bi se obezbedila visoka dostupnost i horizontalna skalabilnost infrastrukture kreiran je *Application Load Balancer* (*ALB*). On kao *target* grupu ima *Auto Scaling* grupu koja, prateći gornje i donje parametre skaliranja, prilagođava broj *EC2* instanci na kojima je hostovan *back-end*.

Za relacione baze podataka koristi se *Amazon RDS* servis. Imajući u vidu da je projekat još uvek u ranoj fazi, nije korišćena *Multi AZ deployment* funkcionalnost kao ni *read* replike.




Slika 1. *Dijagram implementirane infrastrukture*

## 7. DISKUSIJA I ZAKLJUČAK

Primarni cilj rada bio je postavljanje visoko dostupne, skalabilne, fleksibilne kao i bezbedne infrastrukture koja će biti dobra osnova za dalji razvoj širokog spektra veb aplikacija.

Projekat je započet dizajniranjem *VPC* mreže vodeći računa o primeni svih najboljih bezbednosnih praksi. Nakon mreže, fokus je usmeren na dizajniranje i implementaciju resursa potrebnih za hostovanje veb aplikacija. U tu svrhu korišćeni su različiti servisi kao što su *S3, EC2, ALB i CloudFront* radi postizanja skalabilnosti, dostupnosti kao i optimizacije troškova.

Korišćenje *Terraform*-a, kao i samog pristupa u uspostavljanju celokupne infrastrukture putem koda je od presudne važnosti za efikasno upravljanje istom. Infrastrukturu je na ovaj način veoma lako replicirati na više različitih okruženja i izmene postaju veoma jednostavne uz minimiziranje mogućnosti za greške i narušavanje konzistentnosti okruženja.

Ova implementacija uspešno će zadovoljiti potrebe veb aplikacije, kako u početnoj fazi, tako i tokom skaliranja proizvoda. Naravno, prostora za napredak uvek ima. Jedna od potencijalnih oblasti unapređenja jeste korišćenje servisa kao što su *ECS (Elastic Container Service)* ili *EKS (Elastic Kubernetes Service)* za efikasnije upravljanje *docker* kontejnerima i ubrzanje *deployment*-a. Dodatno, uvođenje *CI/CD* pipeline-ova dodatno će poboljšati sistem obezbeđujući kontinuiranu isporuku i integraciju novog koda. Kako bi se greške u implementaciji i potencijalne pretnje otkrile na vreme, preporučuje se korišćenje *CloudWatch* servisa za praćenje logova uz kreiranje *CloudWatch* alarma.

## 8. LITERATURA


[1] AWS Global Infrastructure. https://aws.amazon.com/about-aws/global-infrastructure

[2] The NIST Definition of Cloud Computing. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf

[3] Albert Anthony (2017) - Mastering AWS Security. Create and maintain a secure cloud ecosystem.

[4] Elastic Load Balancing (AWS). https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html

[5] Neil David (2021) - AWS VPC Beginner to Pro - Virtual Private Cloud Tutorial. https://www.youtube.com/watch?v=g2JOHLHh4rI

[6] Amazon Relational Database Service (AWS). https://aws.amazon.com/rds

[7] Mohamed Jawad P (2018) medium.com - Amazon EC2: Auto Scaling. https://medium.com/@jawad846/amazon-ec2-auto-scaling-884ea50d2d

[8] What is Infrastructure as Code (IaC)? https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac

[9] Web Application Firewall (AWS). https://aws.amazon.com/waf

[10] Terraform by HashiCorp. https://www.terraform.io

[11] Angelo Malatacca (2020) medium.com - AWS Terraform S3 and DynamoDB backend https://angelo-malatacca83.medium.com/aws-terraform-s3-and-dynamodb-backend-3b28431a76c1


**Kratka biografija:**


**Denis Fruža** rođen je u Zrenjaninu 1998. Diplomirao je na Fakultetu tehničkih nauka 2021. god. i tada upisuje master studije.

kontakt: denisfruza98@hotmail.com