Aplikacija za beleženje osebne knjižnice

Predmet: Računalništvo v Oblaku in mobilne storitve

Pripravil: Žiga Terzič

Mentor: Dušan Fugina

Somentor: Gregor Zorc

Kazalo vsebine

[Uvod 4](#_Toc197538445)

[VPC 5](#_Toc197538446)

[EC2 6](#_Toc197538447)

[Nastavitev privatne instance in podatkovne baze 8](#_Toc197538448)

[GitHub Actions 10](#_Toc197538449)

[API 12](#_Toc197538450)

[GUI 13](#_Toc197538451)

[Funkcije grafičnega vmesnika 15](#_Toc197538452)

[Prikaz naslovnic 15](#_Toc197538453)

[Dodajanje knjig preko vmesnika 18](#_Toc197538454)

Kazalo Slik

[Slika 1: konfiguracija vpc 3](#_Toc197538367)

[Slika 2: podomrežja 3](#_Toc197538368)

[Slika 3: začasna shema vpc 3](#_Toc197538369)

[Slika 4: končna shema vpc 4](#_Toc197538370)

[Slika 5: SG za javno ec2 instanco 4](#_Toc197538371)

[Slika 6: Sg za db instanco – ssh in db port 5](#_Toc197538372)

[Slika 7: Omrežne nastavitve ec2 instanc 5](#_Toc197538373)

[Slika 8: install mariadb-server 6](#_Toc197538374)

[Slika 9: konfiguracijska datoteka mariadb 7](#_Toc197538375)

[Slika 10: uspešna povezava na db, iz privatne instance 7](#_Toc197538376)

[Slika 11: ustvarjena in polnjena tabela books 8](#_Toc197538377)

[Slika 12: prikaz tabele books 8](#_Toc197538378)

[Slika 13: github actions skrivnosti 9](#_Toc197538379)

[Slika 14: github actions deploy.yml 9](#_Toc197538380)

[Slika 15: config.php 10](#_Toc197538381)

[Slika 16: api/index.php 11](#_Toc197538382)

[Slika 17: api/add.php 11](#_Toc197538383)

[Slika 18:: konfiguracija dostopnih strani nginx strežnika 12](#_Toc197538384)

[Slika 19: zaključek nginx konfiguracije 12](#_Toc197538385)

[Slika 20: Prikaz tabele knjig 13](#_Toc197538386)

[Slika 21: php koda ki populira tabelo 14](#_Toc197538387)

[Slika 22: konfiguracija javnega dostopa do s3 bucketa 15](#_Toc197538388)

[Slika 23: koda obrazca za dodajanje knjig 16](#_Toc197538389)

[Slika 24: nalaganje na s3 16](#_Toc197538390)

[Slika 25: funkcija za povezavo do db in vnos 17](#_Toc197538391)

## Uvod

Za končni projekt pri predmetu Računalništvo v Oblaku in mobilne storitve, smo izbrali razvoj aplikacije za beleženje knjig. Uporabili smo tehnologije AWS, ki smo jih osvojili na predavanjih in vajah:

* VPC – virtualna privatna omrežja
* EC2 – Virtualna mašina
* S3 Bucket – Virtualna hramba datotek

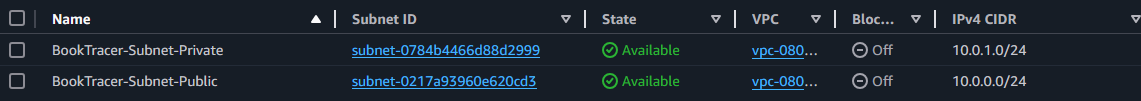
## VPC

A screenshot of a computer

AI-generated content may be incorrect.

Slika 1: konfiguracija vpc

Subnets



Slika 2: podomrežja

Začasna shema – oba podomrežja imata dostop do spleta, za lažje nastavljanje EC2 instanc. Kasneje odstranimo subnet asociacijo za privatno podomrežje, iz usmeritvene zabele.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 3: začasna shema vpc

Ko smo opravili začetno konfiguracijo privatne instance, smo odstranili njen dostop do spleta. A screenshot of a computer

AI-generated content may be incorrect.

Slika 4: končna shema vpc

## EC2

Za vsak tip EC2 instance smo ustvarili varnostno skupino. Javni smo omejili SSH dostop le našemu ip naslovu. Privatni instanci moramo dodatno še definirati dostopni port 3306, za dostop do podatkovne baze.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 5: SG za javno ec2 instanco

A screenshot of a computer

AI-generated content may be incorrect.

Slika 6: Sg za db instanco – ssh in db port

Za ec2 instance smo uporabili sliko debian, in t2.micro konfiguracijo, ki jo nudi »free« nivo aws storitve.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 7: Omrežne nastavitve ec2 instanc

## Nastavitev privatne instance in podatkovne baze

Za prvo nastavitev db instance, smo privatno podomrežje dodali na usmerjevalno tabelo, in se s uporabo PuTTY programa povezali direkno na ec2 instanco. Tam smo naložili mariadb-server s ukazi

sudo apt update

sudo apt install mariadb-server -y

s ukazom »sudo systemctl status mariadb« preverimo stanje aplikacije.

A computer screen with text on it

AI-generated content may be incorrect.

Slika 8: install mariadb-server

S »sudo mariadb« dostopimo do mariadb aplikacije.

Preko mariadb terminala smo ustvarili podatkovno bazo, ter uporabnika s katerim bomo dostopali do baze. Uporabniku smo še dodelili dostope.

Ker mariadb server privzeto posluša le na 127.0.0.1 moramo urediti »bind-adress« naslov v konfiguracijski datoteki 50-server.cnf, s ukazom:

sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

A screenshot of a computer

AI-generated content may be incorrect.

Slika 9: konfiguracijska datoteka mariadb

Do podatkovne baze lahko zdaj dostopamo preko javne ec2 instance, s ukazom:

mysql -h 10.0.1.120 -u ziga -p

kjer je 10.0.1.120 privatni ip naslov privatne ec2 instance

A screenshot of a computer

AI-generated content may be incorrect.

Slika 10: uspešna povezava na db, iz privatne instance

Ustvarili smo tabelo books, s atributi:

* Isbn
* Title
* Author
* Year\_published

Nato smo vstavili nekaj knjig:

A computer screen shot of a black screen

AI-generated content may be incorrect.

Slika 11: ustvarjena in polnjena tabela books

S 'SELECT \* FROM books;' preverimo stanje tabele: A screen shot of a computer

AI-generated content may be incorrect.

Slika 12: prikaz tabele books

## GitHub Actions

Za lažji razvoj smo se odločili, povezati ec2 instanco s github-om, da lahko s uporabo github actions avtomatsko deployamo spremembe. Za ta namen smo ustvarili github repozitorija:

* Zterzicalma/booktracker/gui
* Zterzicalma/booktracker/api

Za uporabo github actions za deploy, smo v obeh repozitorijih dodali podatke za avtomatsko povezavo kot Repository Secrets, kjer so:

* EC2\_host, ip naslov ec2 instance na katero nalagamo
* EC2\_Username, uporabnik ki bo uporabljen v procesu
* EC2\_Private\_key, .pem oblika ključa, za povezavo.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 13: github actions skrivnosti

Nato smo v zavihku »Actions« ustvarili nov »workflow« z imenom deploy.yml, ki je večinoma enak v obeh repozitorijih. Edina razlika je v lokaciji, kjer se izvede git pull ukaz.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 14: github actions deploy.yml

## API

Poverilnice smo zapisali v .env datoteko, ki smo jo dodali v .gitignore.

Config.php prejme poverilnice iz .env in vsebuje funkcijo za povezavo na db.

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 15: config.php

Index.php vsebuje glavno funkcijo API-ja, ki pridobi vse vrstice v tabeli books.

A computer screen shot of text

AI-generated content may be incorrect.

Slika 16: api/index.php

Kasneje smo dodali še funkcionalnost dodajanja knjig preko GUI-ja. To podpira add.php

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 17: api/add.php

## GUI

Za delovanje grafičnega vmesnika, smo najprej naložili php modul

sudo apt install php php-fpm php-mysql -y

nato smo konfigurirali Nginx strežnik, da ta servira PHP za naše datoteke

sudo nano /etc/nginx/sites-available/booktracker

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 18:: konfiguracija dostopnih strani nginx strežnika

ter to omogočili s ukazi

sudo ln -s /etc/nginx/sites-available/booktracker /etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl reload nginx

A screen shot of a computer code

AI-generated content may be incorrect.

Slika 19: zaključek nginx konfiguracije

Na koncu smo odstranili še privzeto omogočeno stran »default«

## Funkcije grafičnega vmesnika

Najprej smo dodali preprosto funkcionalnost klica api-ja ki populira tabelo.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 20: Prikaz tabele knjig

Nato smo tabelo povezali še s S3 bucketom, kamor smo naložili platnice knjig in jih poimenovali [isbn].png. Ko je stran naložena, je poklican api/index.php, ki prejme vsebino podatkovne baze. PHP nato odgovor predela, in sestavi tabelo za vsako vrstico.

### Prikaz naslovnic

Ker smo v s3 bucketu izbrali unikatno imensko shemo isbn, ki je tudi v podatkovni bazi, za prikaz slik knjig, preprosto vnesemo <img src> element s povezavo do slik.

A screen shot of a computer code

AI-generated content may be incorrect.

Slika 21: php koda ki populira tabelo

Da smo lahko dodali funkcionalnost, prikaza slik, ki so naložene na S3 bucket-u, sno najprej morali omogočiti javni dostop do datotek v bucketu. Dodali smo tudi dostopno politiko:

A screenshot of a computer

AI-generated content may be incorrect.

Slika 22: konfiguracija javnega dostopa do s3 bucketa

### Dodajanje knjig preko vmesnika

A computer code on a black background

AI-generated content may be incorrect.

Slika 23: koda obrazca za dodajanje knjig

Ta obrazec podatke preda add\_book.php, ki preveri prisotnost podatkov ustvari povezavo s s3 bucketom ter naloži sliko. Ostale podatke o knjigi v podatkovno bazo vnese preko API-ja v /api/add.php

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 24: nalaganje na s3

A screen shot of a computer program

AI-generated content may be incorrect.

Slika 25: funkcija za povezavo do db in vnos