Usporedba platformi za razvoj aplikacija u oblacima

Knežević, Jakov

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:222372

Rights / Prava: Attribution-NonCommercial-NoDerivs 3.0 Unported/Imenovanje-Nekomercijalno-Bez prerada 3.0

Download date / Datum preuzimanja: **2024-03-28**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jakov Knežević

USPOREDBA PLATFORMI ZA RAZVOJ APLIKACIJA U OBLACIMA

ZAVRŠNI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jakov Knežević

Matični broj: 0016139974

Studij: Primjena informacijske tehnologije u poslovanju

USPOREDBA PLATFORMI ZA RAZVOJ APLIKACIJA U OBLACIMA

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Darko Andročec

Varaždin, kolovoz 2021.

Jakov Knežević

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Ovim završnim radom će se opisati što su podatkovni centri, odnosno što su i koja je razlika između klasičnih podatkovnih centara i podatkovnih centara u računalnim oblacima. Nadalje, biti će objašnjeni tipovi računalnih oblaka, dakle što su javni, privatni i hibridni računalni oblaci. Nakon toga će biti pojašnjeni tipovi servisa u računalnim oblacima, odnosno što su IaaS, PaaS i SaaS te će nakon toga, jer je u ovom završnom radu prema zadanoj temi naglasak na platformama kao servisima, biti uspoređeno nekoliko platformi kao servisa. Usporedba će biti izvršena na način da će se opisati značajke, te neke prednosti i nedostatci nekoliko platformi kao servisa.

Nakon toga će biti pomoću jednostavne aplikacije u React programskom kodu uspoređene dvije platforme na način da se prikaže koliko je postupak podizanja aplikacije na poslužitelj jednostavan ili ne, te kakve mogućnosti pruža svaka od platformi. Poslije usporedbe slijedi zaključak gdje će biti izvedena neka osobna mišljenja o podatkovnim centrima te o prethodno obavljenoj usporedbi platformi.

Sadržaj

1. Uvod 6

2. Metode i tehnike rada 7

3. Odnos klasičnih podatkovnih centara i podatkovnih centara u računalnim oblacima 8

4. Računalstvo u oblacima 14

4.1. Tipovi računalnih oblaka 16

4.1.1. Javni računalni oblak 16

4.1.2. Privatni računalni oblak 17

4.1.3. Hibridni računalni oblak 18

4.2. Tipovi računalstva u oblacima 19

4.2.1. Softver kao usluga (SaaS) 20

4.2.2. Infrastruktura kao usluga (IaaS) 21

4.2.3. Platforma kao usluga (PaaS) 23

5. Usporedba platformi za razvoj aplikacija u oblacima 25

5.1 AWS Elastic Beanstalk 25

5.2 Windows Azure 27

5.3 Google App Engine 28

5.4 IBM Cloud 30

6. Usporedba AWS Elastic Beanstalk-a i Windows Azure-a kao platformi za razvoj aplikacija 32

6.1. Microsoft Azure 32

6.2. AWS Elastic Beanstalk 38

7. Zaključak 47

8. Popis literature 49

9. Popis slika 51

10. Popis tablica 52

1. Uvod

Tema završnog rada je “Usporedba platformi za razvoj aplikacija u oblacima” te će time biti obrađena osnovna teorija o klasičnim podatkovnim centrima kao i o podatkovnim centrima u oblaku. Oba tipa podatkovnih centara će biti ukratko uspoređena kako bi imali neku osnovnu ideju o tome kako se ponaša i što je zapravo jedan i/ili drugi tip podatkovnog centra. No osnovna razlika ova dva tipa je što kod klasičnog podatkovnog centra, mi moramo sami nabaviti i financirati svu opremu, infrastrukturu, radnike i ostalo, dok kod podatkovnog centra u oblaku mi od poslužitelja ovih podatkovnih centara pozajmljujemo sve navedeno te time troškove smanjujemo drastično te je jedino što plaćamo vezano uz podatkovni centar poslužiteljeve usluge koje nam pruža.

Nadalje, biti će opisani i obrađeni svi tipovi servisa, dakle SaaS, PaaS te IaaS te će biti navedeni neki primjeri poslužitelja tih servisa. S obzirom da je tema usporedba platformi, od svih tipova servisa biti će najviše obrađivan i detaljnije opisan PaaS jer je on “platforma kao servis”. On pruža brzu izgradnju aplikacija te za poznatije poslužitelje PaaS-a postoji dosta dokumentacije na Internetu što još olakšava ovaj proces.

Jednako tako, biti će obrađene detaljnije, odnosno uspoređene, dvije platforme kao servis, AWS Elastic Beanstalk i Windows Azure. Biti će uspoređene po procesu izgradnje aplikacije te koliko je bio taj proces jednostavan ili ne čime ćemo na kraju dobiti “pobjednika” ove usporedbe.

2. Metode i tehnike rada

Informacije na temu ovog završnog rada će biti dobivene putem Interneta jer je putem njega najlakše, najbrže i najjednostavnije doći do informacija. Izvori će naravno biti navedeni pri kraju završnog rada, a biti će korištene informacije iz raznih članaka te sa službenih web stranica tvrtki koje pružaju navedene servise u završnom radu.

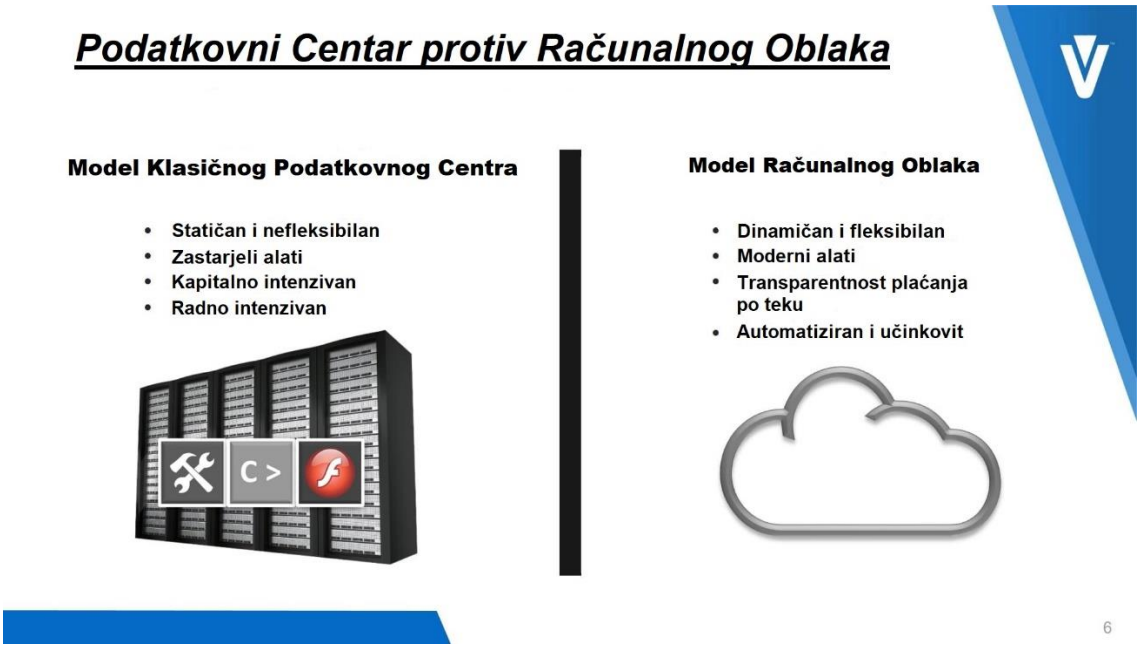
Što se tiče teme, odnosno usporedbe platformi za razvoj aplikacija u oblacima, biti će najviše obrađene dvije platforme kao servis, a to su AWS Elastic Beanstalk i Windows Azure iz razloga što su se one činile kao najbolja opcija prilikom istraživanja o platformama kao servisima. Obje platforme kao servis su dostupne putem Interneta te svaka od njih ima besplatno probno trajanje što će naravno i biti iskorišteno u svrhu istraživanja i usporedbe ove dvije platforme.

AWS Elastic Beanstalk je dakle Amazon-ova verzija PaaS-a te na prvu izgleda vrlo interaktivno te time ima dojam da se njome lako koristi. Većina operacija se odvija putem Amazon-ove web stranice, odnosno sučelja, te za jednostavnije web aplikacije nemamo potrebu ništa preuzimati kako bismo naš izvorni kod učitali na njihovu platformu.

Windows Azure na prvu nije izgledao jednako interaktivno kao AWS Elastic Beanstalk, no na kraju se ispostavilo da je bilo relativno lako postaviti sve potrebno za, u ovom slučaju, web aplikaciju. Jednako kao i kod AWS Elastic Beanstalk-a, većina se operacija izvodi putem Microsoft Azure portala, dok se neki dijelovi moraju izvoditi putem Azure CLI-ja, što je zapravo naredbeni redak.

3. Odnos klasičnih podatkovnih centara i podatkovnih centara u računalnim oblacima

Klasični podatkovni centri su zapravo fizička mjesta u kojima kompanije, poduzeća i slični pohranjuju podatke i aplikacije neophodne za njihov rad. Dakle, unutar klasičnog podatkovnog centra može postojati sve od rutera, sigurnosnih uređaja pa do sustava za pohranu i kontrolera za isporuku aplikacija. Uz to, kako bi sva ova oprema i njen potreban softver bio ažuran, klasični podatkovni centar mora imati veliku količinu infrastrukture, a neki od dijelova uključuju ventilaciju i rashladne sustave, neprekidne izvore napajanja što uključuje i rezervne generatore [11].



Slika 1: Klasični podatkovni centar vs. podatkovni centar u oblaku, preuzeto s: https://twitter.com/OracleIaaS/status/974450207785185281/photo/1

U odnosu na njega, podatkovni centri koji se nalaze u računalnim oblacima su podosta različiti po svom sastavu. Ovakvi podatkovni centri podatke spremaju ne lokalno, odnosno fizički, već na Internetu na način da kada mi spremimo naše podatke na jedan od takvih podatkovnih centara, oni se automatski fragmentiraju i poduplavaju te se spremaju na više lokacija kako bi bili što sigurniji. Nadalje, u slučaju nepogode, klasični

podatkovni centri su u problemu, dok podatkovni centri u računalnim oblacima imaju spremljene sigurnosne kopije podataka, a često se znaju izrađivati i sigurnosne kopije sigurnosnih kopija čime se maksimizira očuvanost podataka [11].

Dalje, što se tiče cijene, ako želimo sagraditi klasični podatkovni centar, moramo kupiti razne komponente koje uključuju hardver za server te mrežu uz što naravno dolazi i njegovo obnavljanje, odnosno kupovanje novog svakih nekoliko godina te moramo imati i stručno obrazovano osoblje koje će znati rukovati, upravljati te održavati ovaj hardver. S druge strane, kada podatke spremimo u podatkovni centar na računalnom oblaku, zapravo koristimo tuđi hardver i infrastrukturu čime mnogo uštedimo na tim resursima kao i na stručnom osoblju koje je zaduženo za njegovo održavanje, rukovanje te ostalo što nam omogućuje da neiskorišteni novac usmjerimo negdje drugdje [11].

Nadalje, gledano s aspekta pristupačnosti, klasični podatkovni centar nam pruža mogućnost fleksibilnosti u obliku izbora opreme koju želimo koristiti čime ćemo znati što stvarno koristimo. Ovo je pozitivna stvar jer ukoliko poželimo zamijeniti neki dio opreme ili ga unaprijediti, možemo to učiniti bez problema jer je sva oprema naša te nam nitko “ne stoji na putu” kako bismo mogli realizirati željene promjene. S druge strane, podatkovni centar u računalnom oblaku nam ne može pružiti ovakvu fleksibilnost iz razloga što oprema nije naša, a uz to ne znamo ni koja se oprema točno koristi čime nam se gubi mogućnost izmjene željenih dijelova opreme zbog čega moramo kontaktirati i dogovoriti se s poslužiteljem podatkovnog centra u računalom oblaku kako bismo postigli nekakav sporazum. Također, podatkovni centar u računalnom oblaku funkcionira putem, odnosno pomoću Interneta što stvara potencijalni problem ukoliko nam nestane Internetska veza. Ukoliko dođe do takvog problema, privremeno ćemo biti odsječeni od naših podataka koji se nalaze u računalnom oblaku našeg poslužitelja što nam može stvoriti problem pri obavljanju posla te narušava kvalitetu, neprekidnost i ostale aspekte isporuke usluge/proizvoda. Nasreću, danas je ovaj tip problema rijedak te on ne bi trebao previše zabrinjavati one koji se žele prebaciti i/ili prijeći na podatkovni centar u računalnom oblaku [11].

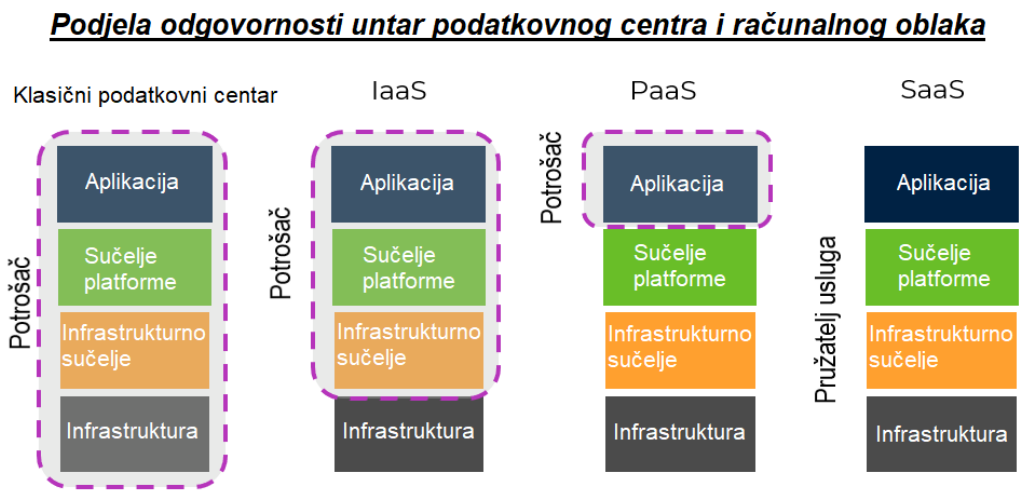
Sigurnost je naravno još jedan od iznimno bitnih aspekata ove priče jer je sigurnost i integritet naših podataka iznimno bitan dio poslovanja, kako iz pogleda imidža potrošača i poslužitelja, tako i iz pogleda odnosa potrošača i poslužitelja. Kod klasičnog podatkovnog centra, sigurnost je u obliku zapošljavanja zaštitara koji će biti zaduženi za očuvanje podatkovnog centra i njegovih podataka. Ovo ima svoje prednosti do neke mjere na način da imamo potpunu kontrolu nad našom opremom te će ujedno podacima moći pristupiti samo osobe od povjerenja. Ako pogledamo na sigurnost u podatkovnom centru u računalnom oblaku, sigurnost je zajamčena na način da će se naš poslužitelj potruditi da svim mjerama i mogućnostima zaštiti naše podatke od krađe ili petljanja s našim podacima. Za ovaj zadatak je odgovorno stručno osoblje koje ima za zadatak primijeniti sigurnosne mjere kojima će moći uspješno zaštiti naše podatke te zaustaviti hakere u njihovom naumu [11].

Zadnji aspekt koji je bitan za razmotriti je skalabilnost podatkovnih centara. Klasični podatkovni centar moramo, kao što je i ranije navedeno, graditi od nule što nam odmah govori kako ćemo imati visoke troškove izgradnje, kupovine opreme, njenog postavljanja, održavanja, plaćanja stručnog osoblja koje će njime rukovati te ostali troškovi koji se mogu brzo nagomilati. Nadalje, ukoliko nam se ukaže potreba za proširenjem naših kapaciteta, morati ćemo ponovno izdvojiti određenu sumu novaca kako bismo to mogli realizirati. Iz ovoga možemo vidjeti kako je za klasični podatkovni centar potrebno imati povelik kapital kako bismo mogli izvršavati izmjene koje trebamo i želimo što naravno stvara problem ukoliko ne posjedujemo takav kapital. Podatkovni centar u računalnom oblaku ne zahtjeva takav ogroman kapital jer nemamo potrebe plaćati opremu, stručno osoblje, izgradnju samog podatkovnog centra i ostalog jer je za to sve zadužen naš poslužitelj koji nam sve to pruža za određenu cijenu. Također, bitno je napomenuti kako u slučaju da imamo potrebu proširiti naš podatkovni centar, poslužitelji su veoma fleksibilni te nam mogu pružiti razne opcije za proširenje kojima nam se prilagođavaju kako bi zadovoljili naše potrebe. Naravno, ukoliko su želje potrošača obratne, te on želi smanjiti svoje troškove na način da smanji količinu prostora za pohranu koji plaća, a ne iskorištava u potpunosti, poslužitelji su jednako tako spremni ponuditi

opcije i za takve odluke. Nakon što smo prošli ove sve aspekte jednog i drugog tipa podatkovnog centra možemo izvesti neke njihove pozitivne i negativne strane [11].

Pozitivne strane klasičnog podatkovnog centra su kontrola koju imamo nad opremom, stručnim osobljem, mogućnošću izmjene opreme koju koristimo. Dakle, ako želimo izmijeniti neki dio ili ga unaprijediti, možemo to slobodno učiniti jer znamo točno koju opremu koristimo te koliko je ona stara i je li ju potrebno izmijeniti, odnosno unaprijediti. Druga pozitivna strana klasičnih podatkovnih centara je što u slučaju nestanka Internetske veze, mi i dalje imamo pristup našim podacima što je velika prednost jer i dalje možemo normalno raditi s našim podacima bez brige. Dalje, pozitivna strana je i što nam hakeri nisu prijetnja jer naši podaci nisu na Internetu nego su lokalno spremljeni te ih čuvaju zaštitari zaduženi za sigurnost našeg podatkovnog centra.

Negativne strane ovog tipa podatkovnog centra su kao prvo visoka cijena izgradnje, održavanja, plaćanja radnika i sličnog te nam je uz to potreban visok kapital kako bismo mogli sve navedeno realizirati, a uz to nam je potreban kapital i ako želimo proširiti naš podatkovni centar kako bismo imali više resursa i ostalog potrebnog kako bismo mogli uspješno poslovati u punom kapacitetu. Nadalje, za klasični podatkovni centar je potrebno zaposliti stručno osoblje koje će znati kvalitetno i pravilno rukovati, upravljati te održavati naš podatkovni centar jer ukoliko u ovom dijelu dođe do problema, naše cijelo poslovanje je u problemu i riziku od gubitka potrošača. Jednako tako, moramo se i zaštiti od problema kao što su prirodne nepogode ili problemi u obliku nestanka napajanja zbog čega moramo imati i rezervne generatore kako bismo imali neprekidan izvor napajanja, a u slučaju prirodnih nepogoda moramo imati i spremljene ažurne, redovite i na sigurnom mjestu spremljene sigurnosne kopije podataka naših korisnika.



Slika 2: Podjela odgovornosti korisnika različitih tipova podatkovnih centara, preuzeto s: https://info.veritasts.com/hs-fs/hubfs/DR%20in%20cloud%20image.png?width=987&name=DR%20in%20cloud%20image.png

Podatkovni centar u računalnom oblaku ima pozitivne strane u obliku da cijena nije ni blizu tolik problem jer kao prvo, ne moramo misliti o plaćanju izgradnje podatkovnog centra, plaćanju stručnog osoblja, zaštitara, cijele infrastrukture, opreme i sličnog jer se o tome svemu brine poslužitelj koji nam je za određenu cijenu pružio pristup svojim resursima i mjestom za pohranu podataka čime velik dio kapitala koji bi kod drugog tipa podatkovnog centra morali uložiti u sve ranije navedeno, možemo uložiti u druge svrhe kao što može biti recimo marketing naših proizvoda i/ili usluga. Dalje, što se tiče prirodnih nepogoda ili sličnog smo osigurani od strane poslužitelja koji je dužan pobrinuti se za očuvanje naših podataka koji su spremljeni u više sigurnosnih kopija. Skalabilnost ovog tipa podatkovnog centra je puno bolja od drugog tipa jer ne moramo misliti o tome koliko ćemo morati platiti proširenje i novu opremu kako bi se proširili već o tome vodi brigu poslužitelj koji već ima pretkreirane planove ukoliko se neki od potrošača žele proširiti ili zahtijevaju više resursa ili mjesta za pohranu.

Negativne strane ovog tipa podatkovnog centra su potencijalno sigurnost iz razloga što su naši podaci spremljeni na Internetu te su time izloženi napadima hakera koji žele ukrasti podatke, petljati s njima ili slično. Iako je sigurnost u negativnim stranama, u većini slučajeva poslužitelji imaju stručno osoblje koje je zaduženo nositi se takvim situacijama te očuvanjem sigurnosti i integriteta podataka kako bi hakerima što više

otežali “posao”. S obzirom da je ovo tip podatkovnog centra kod kojeg su podaci spremljeni na Internetu, možemo vidjeti koji se problem može pojaviti, a to je nestanak Internetske veze. Ovo može biti veoma problematično iz razloga što nam to može ozbiljno narušiti poslovanje jer nemamo pristup našim podacima te nam to može narušiti imidž te neprekidnost i kvalitetu proizvoda i/ili usluge ukoliko se događa često. Danas je ovo relativno rijedak problem jer se kompanije često osiguraju od ovog problema na neki način što čini ovaj problem manjim. Nadalje, još neka od potencijalno negativnih strana podatkovnog centra u računalnom oblaku je što potrošač nema potpunu kontrolu nad opremom koju koristi jer koristi tuđu opremu koju ne može mijenjati po svojoj volji i želji što nekim potrošačima može predstavljati problem.

4. Računalstvo u oblacima

Računalstvo u oblacima je zapravo isporuka računalnih usluga na zahtjev u što spada sve od aplikacija do mjesta za pohranu, pa čak i sve do procesorske snage. Dakle, sve ove značajke su spremne za korištenje uobičajeno preko Interneta te na bazi protočne sheme, odnosno to je način plaćanja kod kojeg privatna osoba ili organizacija plaća troškove kupljenog proizvoda i/ili usluge u trenutku kada ih one zatrebaju, a ne prije ili poslije korištenja ili potrebe [8].

Temeljni koncept koji se krije iza računalstva oblacima je da je lokacija servisa te ostalih detalja kao što je hardver, operacijski sustav i slično potpuno nebitno krajnjem korisniku. Nadalje, ovaj koncept je postojao već dugi niz godina, čak još u 1960-ima kada je ovaj koncept imao ideju da računalni biroi “posuđuju” kompanijama pristup glavnom računalu umjesto da ga one same kupuju. No ovaj koncept nije živio dugo jer se brzo počeo razvijati koncept PC-ja, odnosno osobnog računala, čime je postalo jeftinije kupiti osobno računalo te je samim time počeo rasti koncept podatkovnih centara gdje bi kompanije, kojima je sada jeftinije posjedovati računala, spremale velike količine podataka. Danas, računalstvo u oblacima je buknulo te je postalo veoma bitan aspekt poslovanja, a jedan od većih poslužitelja servisa u računalnim oblacima je Amazon Web Services [8].

Način na koji računalstvo u oblacima zapravo funkcionira je sljedeći, umjesto da smo mi vlasnici, tj. da mi kupujemo računalnu infrastrukturu, podatkovne centre ili slično, možemo sve navedeno iznajmiti od poslužitelja računalnih usluga. Velika prednost korištenja usluga baziranima na servisima u računalnim oblacima je ta što kompanije ili manje firme “izbjegavaju" plaćanje unaprijednih troškova kao i samu kompleksnost posjedovanja te održavanja vlastite IT (eng. “Information Technology") infrastrukture. Ovo zapravo znači da oni plaćaju ono što koriste i kada koriste dok zauzvrat, poslužitelji ovih računalnih servisa u oblacima imaju koristi u obliku velikog udjela na tržištu na način što pružaju jednake usluge velikom broju potrošača. Kako su se vremenom usluge u

računalnim oblacima unaprjeđivale, tako se i asortiman njihovih usluga povećavao. Danas, njihov asortiman usluga seže od nekih osnovnih, kao što su pohrana podataka, mrežne usluge, procesorska snaga te neke osnovne uredske aplikacije pa sve do nekih kompleksnijih usluga kao što su procesiranje prirodnog govora pomoću računala te umjetna inteligencija [8].

Neki od primjera računalnih oblaka su Gmail, dakle globalno poznati servis za elektroničku poštu, sigurnosno kopiranje koje se sprema u računalnom oblaku u unutar kojeg mogu biti pohranjeni razni mediji, veoma je popularno danas spremanje fotografija i/ili video zapisa sa naših pametnih telefona na računalni oblak. Osim ovoga, imamo i primjer servisa koji omogućavaju velikim kompanijama da sve svoje podatke i aplikacije imaju pohranjene u računalnim oblacima te ih od tamo i pokreću. Nadalje, svjetski poznati Netflix je primjer usluge koja koristi usluge računalnog oblaka kako bi bio u mogućnosti pokretati svoje servisa za pokretanje videozapisa, odnosno filmova, TV serija i sličnog te ostalih usluga koje on pruža [8].

Računalni oblak je iznimno bitan što potkrepljuje činjenica da je trećina troškova na globalnoj razini vezano uz izgradnju infrastrukture koja ima mogućnost podržavanja računalnih oblaka. Nadalje, bitno je napomenuti kako se predviđa da će ove godine oko trećine troškova IT poduzeća biti odijeljeno za poslužitelje računalnih oblaka te njihove usluge što je naravno indikator da se poduzeća, kompanije te ostali sve više oslanjaju na računalne oblake te usluge koje oni pružaju [8].

4.1. Tipovi računalnih oblaka

4.1.1. Javni računalni oblak

Javni oblak je klasični oblik, odnosno model računalnog oblaka kod kojeg korisnici imaju lak pristup računalnim resursima, bilo to putem IaaS-a, PaaS-a ili SaaS-a. On može biti besplatan, a može biti za prodaju na zahtjev pri čemu potrošači plaćaju po CPU ciklusu, iskorištenoj pohrani podataka ili količini propusnosti koje potroše. Prednost javnog oblaka je njegova sposobnost brzog podizanja nekog servisa na veliku razinu. Poslužitelji računalnih oblaka imaju ogromne količine računalnih resursa koje dijele među stotinama ili tisućama potrošača što se naziva višenamjenska arhitektura. Nadalje, ovakvi poslužitelji imaju toliko računalnih resursa da imaju i viška resursa koje mogu po potrebi dodijeliti nekom od potrošača ukoliko im se ukaže potreba za više resursa. Zbog ovoga se ovaj tip računalnog oblaka najviše koristi za aplikacije koje su manje osjetljive, a da pritom zahtijevaju varirajuće količine računalnih resursa [8].

Za razliku od privatnih računalnih oblaka, objašnjenih u nastavku, javni računalni oblaci mogu spasiti kompanije od troškova za kupovinu, postavljanje i održavanje hardvera te aplikacijske infrastrukture jer su za sve navedeno zaduženi poslužitelji računalnih oblaka. Također, javni računalni oblaci se brže postavljaju nego privatni računalni oblaci te kao što je rečeno ranije, imaju daleko veće mogućnosti skaliranja. Na primjer, svaki zaposlenik u nekoj tvrtki može pristupiti istoj aplikaciji iz bilo kojeg ureda ili grane s uređajem po njihovom izboru dokle god imaju pristup Internetskoj vezi. Što se tiče sigurnosti javnog računalnog oblaka, bilo je prijavljenih incidenata, no bez obzira na to, ukoliko se pravilno izvrši implementacija javnog računalnog oblaka, on može biti jednako siguran kao i privatni računalni oblak, pogotovo ako se uz pravilnu implementaciju, koriste i pravilne sigurnosne mjere kao što su sustavi otkrivanja i sprječavanje upada [8].

4.1.2. Privatni računalni oblak

Privatni računalni oblak je tip računalnog oblaka koji omogućava organizacijama da imaju neke od prednosti javnog računalnog oblaka dok pritom ne moraju misliti o kontroli i sigurnosti nad podatcima i servisima, jer se nalaze iza tvrtkinog vatrenog zida čime imaju pregled nad tokom podataka te mogu organizirati i izgraditi infrastrukturu onako kako oni žele. Ovime se sprječava pristup trećoj strani koja nije dio tvrtke da ima uvid u osjetljive podatke i operacije koje se odvijaju unutar tvrtke. Ovaj oblik servisa se pruža najčešće preko Interneta ili unutarnje privatne mreže te samo za odabrane korisnike umjesto za javnost. Najčešće se ovaj način koristi za IaaS ili PaaS projekte kako bi se developerima pružio pristup računalnim resursima koje si dodjeljuju po potrebi na zahtjev bez rizika za sigurnost podataka ili sličnog [8].

Problem kod privatnih oblaka je što ta dodatna sigurnost ima svoju cijenu, a ona je što je malen broj kompanija na tako velikoj razini kao što su AWS, Microsoft ili Google što znači da nisu u stanju kreirati jednaku ekonomiju te razine. Prema svemu navedenom, privatni računalni oblak je dobra osnova za kompanije kako bi pravilno i kvalitetno shvatili kako računalni oblaci funkcioniraju kako bi mogli uspješno i sa dobrom ostvarenom sigurnošću prebaciti svoje poslovanje na javni računalni oblak. Nadalje, još jedna od potencijalno negativnih strana je što tvrtkin IT odjel ima na sebi odgovornost za troškove te održavanje privatnog računalnog oblaka. Prema ovome možemo zaključiti kako privatni oblaci imaju jednake potrebne troškove za osoblje, menadžment te održavanje kao što ih ima i klasični podatkovni centar [8].

Modeli koji se mogu pružiti ukoliko koristimo privatni računalni oblak su IaaS (Infrastruktura kao usluga) koja omogućava tvrtki da koristi infrastrukturne resurse kao što su mreža, računalni resursi ili podatkovna pohrana u obliku usluge. Drugi model je PaaS (Platforma kao usluga) koja pruža tvrtki da može pohraniti bilo što, od nekih jednostavnih aplikacija baziranih na računalnim oblacima do nekih složenijih i kompliciranijih poslovnih aplikacija. Nadalje, postoji način da se privatnih i javni računalni oblik kombiniraju čime nastaje hibridni računalni oblik koji omogućava poslovanju da

imaju prednost skalabilnosti i potreba za resursima na zahtjev kako im zahtjevi za resursima porastu [8].

4.1.3. Hibridni računalni oblak

Treći tip računalnog oblaka je najlakše objasniti na način da “uzima po malo” od prvog i drugog tipa računalnog oblaka čime se tvori kombinacija te dvije vrste koja se naziva hibridni računalni oblak. Dakle, korisnici, odnosno kompanije ili poduzeća žele koristiti jedan ili više javnih računalnih oblaka koji su povezani s privatnim računalnim oblakom ili s klasičnom infrastrukturom podatkovnog centra unutar kompanije ili sličnog [8] [9] [10].

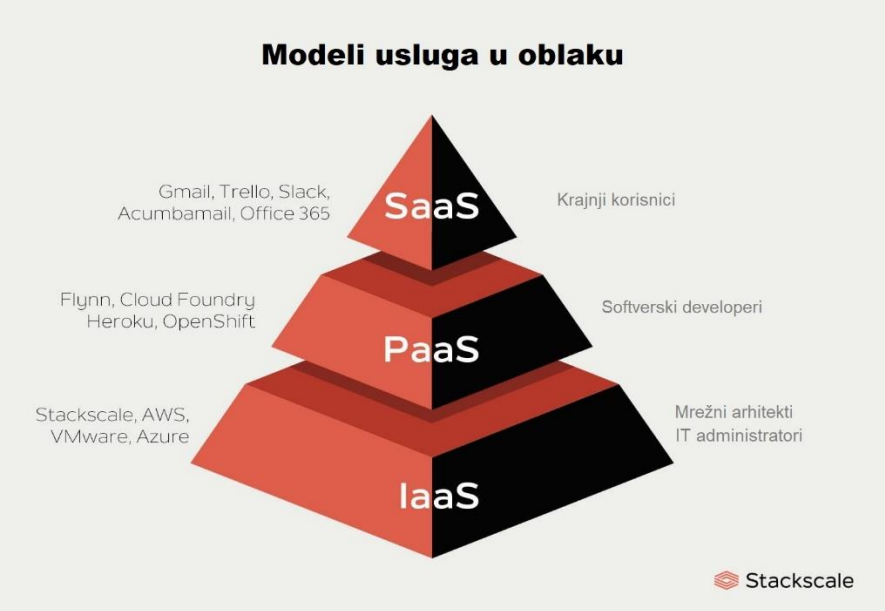
Na primjer, dio podataka imamo pohranjen u javnom računalnom oblaku, a dio u privatnom računalnom oblaku te uz ovo služimo se s nekolicinom dobavljača te različitim razinama korištenja računalnog oblaka. Privatni i javni računalni oblak rade neovisno jedan o drugom, a komuniciraju putem kriptirane veze koristeći tehnologiju koja omogućava prenosivost podataka i aplikacija. Bitno je bilo napomenuti da su privatni i javni računalni oblak neovisni jedan o drugom jer to omogućava kompanijama da pohrane zaštićene ili povlaštene podatke na privatnom računalnom oblaku, a s druge strane imaju resurse iz javnog računalnog oblaka pomoću kojih pokreću aplikacije koje se oslanjaju na ove resurse. Ovime se postiže da podaci budu minimalno izloženi jer se ti osjetljivi podaci ne spremaju dugoročno na javni računalni oblak [8] [9] [10].

Glavni razlog zbog čega se kompanije, manje tvrtke ili poduzeća odlučuju za hibridni tip računalnog oblaka je zbog planiranja za slučaj nekakve nepogode i zbog izbjegavanja troška za dodatni hardver u slučaju da se želi proširiti postojeći podatkovni centar. Druge pozitivne strane hibridnog računalnog oblaka je bolja kontrola donositelja IT odluka nad obje komponente, odnosno nad privatnim i javnim računalnim oblakom, u odnosu na korištenje unaprijed namještene platforme javnog računalnog oblaka. Ovime se postiže veća efikasnost te fleksibilnost kako bi se zadovoljile potrebe. Nadalje, još

jedna od pozitivnih strana ovoga je prednost plaćanja za dodatne resurse samo onda kada su nam ti resursi potrebni u odnosu da ih dodatno plaćamo iskoristili mi njih ili ne [8] [9] [10].

Iako hibridni računalni oblak ima svoje prednosti nad korištenjem samo javnog računalnog oblaka, i dalje pati od istih problema, a oni su privatnost te sigurnost. Zbog ovoga mnoge organizacije ili slični odustaju od hibridnog računalnog oblaka jer im su im ti problemi nepotreban i/ili neprihvatljiv sigurnosni rizik. Nadalje, imamo problem u slučaju kada nam je veoma bitan prijenos podataka te latencija jer ako nam je bitno da određeni podaci stignu točno u određeno vrijeme, latencija može izazvati nepoželjne probleme. Novčani problem je također jedan od problema hibridnog računalnog oblaka jer su troškovi servera privatnog dijela visoki što može stvoriti problem ukoliko je budžet relativno malen zbog čega se manjim biznisima bolje isplati koristiti javni računalni oblak kod jednog od takvih poslužitelja [8] [9] [10].

4.2. Tipovi računalstva u oblacima



Slika 3: Piramida tipova računalstva u oblacima, preuzeto s: https://www.stackscale.com/wp-content/uploads/2020/04/cloud-service-models-iaas-paas-saas-stackscale.jpg

4.2.1. Softver kao usluga (SaaS)

Softver kao usluga je način dostavljanja aplikacija na Internetu u obliku usluge s kojom se većina ljudi susreće na dnevnoj bazi. Drugim riječima, umjesto preuzimanja, instaliranja i održavanja softvera možemo im jednostavno pristupiti putem Interneta, izbjegavajući kompleksno softversko i hardversko upravljanje. SaaS aplikacije, često zvane i aplikacije bazirane na Webu, softveri na zahtjev ili softveri poslužitelji. Kako god ih nazvali, svaka aplikacija bazirana na SaaS-u se pokreće pomoću poslužiteljevih servera. Poslužitelj omogućuje pristup aplikacijama uz maksimalnu sigurnost, dostupnost i sve potrebne značajke. Korištenje SaaS aplikacija vrlo je lako i jednostavno, sve što je nužno da bismo im pristupili je internetska veza [1] [2] [3] [4].

SaaS model korisnicima omogućuje privatnost, istovremeno pružajući uslugu koja je pouzdana i vjerodostojna te zahvaljujući tome korisnici nemaju brige o tome hoće li netko pristupiti njihovim osobnim podatcima bez autorizacije [1] [2] [3] [4].

Kod SaaS modela svi korisnici i aplikacije dijele jednu zajedničku infrastrukturu i osnovni kod, a zahvaljujući tome pružatelji usluga mogu puno brže i sigurnije uvoditi inovacije te sačuvati dragocjeno vrijeme koje su prije trošili na održavanje bezbrojnih verzija zastarjelih kodova [1] [2] [3] [4].

SaaS model ima značajku lake i brze prilagodbe aplikacija potrebama svakog korisnika bez promjena u uobičajenoj infrastrukturi i upravo je zbog načina na koji funkcionira izrazito zahvalan za korištenje. Naime, prilagode su jedinstvene za svakog korisnika i tvrtku te uvijek ostaju sačuvane putem nadogradnji. Drugim riječima, pružatelji usluga SaaS-a mogu raditi nadogradnje puno češće uz značajno niže troškove te puno manji rizik za korisnike [1] [2] [3] [4].

Još jedna od značajki SaaS-a je bolji i napredniji pristup podacima s bilo kojeg uređaja povezanog na internet te omogućavanje da svaki korisnik vidi istu informaciju u isto vrijeme [1] [2] [3] [4].

SaaS je 2017. godine bio dominantna vrsta računalstva u oblacima zaslužan za dvije trećine cjelokupnog javnog računalstva u oblacima, a predviđanja su da će 2021. njegov udio pasti malo ispod 60% [1] [2] [3] [4].

SaaS potrošnja sačinjena je od aplikacija i infrastrukturnog softvera za sustav pa je tako tijekom 2019. godine kupovina aplikacija činila više od polovine ukupne javne potrošnje u računalnim oblacima. Aplikacije koje služe upravljanju odnosima s klijentima (CRM) te aplikacije za upravljanje resursima poduzeća (ERM) predstavljaju više od 60% sveukupnih transakcija među aplikacijama u računalnim oblacima u 2021. Godini. Trenutno se radi na razvoju SaaS integracijskih platformi (SIPs) za izgradnju dodatnih SaaS aplikacija. To se još zove “treći val” u softverskom usvajanju [1] [2] [3] [4].

Neki od primjera SaaS aplikacija su: Microsoft Office, Netflix, BigCommerce, Google Apps, Salesforce, Dropbox, MailChimp, ZenDesk, DocuSign, Slack, Hubspot [1] [2] [3] [4].

4.2.2. Infrastruktura kao usluga (IaaS)

Infrastruktura kao usluga je vrsta računalstva u oblacima koja se obično dostavlja organizaciji putem kontrolne ploče ili API-ja, dajući korisnicima potpunu kontrolu nad kompletnom infrastrukturom. IaaS ima istu tehnologiju i mogućnosti kao i klasična baza podatka, ali se ipak razlikuju u činjenici da se ničim od toga ne treba fizički upravljati niti se bilo što treba održavati. Korisnici IaaS-a i dalje mogu direktno pristupiti svojim serverima i pohrani, ali je taj posao primarno prepušten vanjskim tvrtkama putem „virtualnog podatkovnog centra“ u oblaku [3].

U usporedbi s Saas-om i PaaS-om, IaaS- ovi korisnici odgovorni su za upravljanje aplikacijama, podatke, operacijske sustave, vrijeme izvođenja te međuopremu. Kako god, davatelji usluga IaaS-a rukovode tvrdim diskovima, serverima, pohranom, virtualizacijom i umrežavanjem [3].

Karakteristike IaaS-a podrazumijevaju resurse dostupne kao uslugu, dinamičnost i fleksibilnost, cijenu koja varira ovisno o potrošnji, više korisnika koji koriste jedan hardver te organizaciju koja zadržava potpunu kontrolu nad infrastrukturom [3].

U određenim situacijama IaaS se pokazao kao najbolje moguće rješenje zbog svoje fleksibilnosti i prilagodljivosti. Tako je na primjer vrlo koristan za male tvrtke i one koji tek pokreću biznis jer osigurava više vremena i manje potrošnje novca na kreiranje hardvera i softvera. Osim toga, mogu ga koristiti i veće tvrtke koje žele zadržati kompletnu kontrolu nad svojom infrastrukturom i aplikacijama, a žele nabaviti samo ono što im je stvarno potrebno. Također, tvrtkama koje žele ostvariti velik dobitak pogoduje skalabilnost, odnosno fleksibilnost IaaS-a jer lako mogu promijeniti hardver i softver ovisno o razvoju njihovih potreba [3].

Prednosti IaaS osim već spomenutih fleksibilnosti i prilagodljivosti su i jednostavna automatizacija prostora za pohranu, umrežavanja i servera, kupnja hardvera koja se može bazirati na potrošnji, kupnja resursa prema potrebu te mogućnost da klijenti zadrže potpunu kontrolu nad svojom infrastrukturom [3].

Nadalje, nedostaci IaaS-a vezani su za sigurnost jer korisnik kontrolira sve podatke, aplikacije, OS platformu te posrednički softver zbog čega može doći do sigurnosnih prijetnji od strane poslužitelja ili drugih strana. Tako može doći do propusta u obliku da neovlašteni subjekti dođu do podataka. Osim toga, hardverski resursi se dinamički raspoređuju među korisnicima kako su dostupni pa se tako dobavljač mora pobrinuti da ostali korisnici ne mogu pristupiti podacima koje su prethodni korisnici pohranili [3].

Neki od primjera IaaS-a su: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, OpenShift... [3]

4.2.3. Platforma kao usluga (PaaS)

Platforma kao usluga je tip računalstva u računalnom oblaku koji ima model isporuke sličan kao i kod SaaS-a, ali se razlikuju u činjenici što PaaS umjesto isporuke softvera putem interneta pruža platformu za izradu softvera. Ta platforma se isporučuje preko mreže što programerima omogućuje da se fokusiraju na izradu softvera bez brige oko pohrane, operativnih sustava, ažuriranja softvera te infrastrukture [3] [5] [6] [7].

PaaS platforme su dostupne velikom broju korisnika putem iste razvojne aplikacije, mogu se nadograđivati na tehnologiji virtualizacije pa se tako resursi mogu lako smanjivati ili povećavati ovisno o tome kako se poslovanje mijenja. Također, pruža mnoštvo usluga korisnik za razvoj i testiranje aplikacija te integrira web usluge i bazu podataka, a uz to lako se pokreće bez prevelikog znanja o administraciji sustava [3] [5] [6] [7].

PaaS je često ekonomski i vremenski najisplativiji način za programera koji želi brzo razviti ili postaviti jedinstvenu i prilagođenu aplikaciju. Naime, idealan je jer omogućuje programeru da se bavi razvojem kreativne strane aplikacije umjesto da se bavi ažuriranjima softvera ili sigurnosnim zakrpama. Osim toga, ovaj tip računalstva u oblacima može pojednostaviti posao ako više programera radi na istom projektu te pruža brzinu i fleksibilnost ako se u projekt uključe i drugi dobavljači [3] [5] [6] [7].

Prednosti kod PaaS-a su laka dostupnost, značajno smanjenje potrebe za pisanjem opsežnog koda te lak prijelaz na hibridni model. Vrlo je čest i popularan izbor za tvrtke koje žele razviti i kreirati jedinstvene aplikacije bez trošenja mnoštva novca ili preuzimanja velikih odgovornosti. Kod korištenja PaaS-a sve što je potrebno nalazi se na

platformi, odnosno operativni sustav, vatrozid, sigurnosni sustav te poslužiteljski softver. Za postavljanje bilo čega od navedenog nije potrebno platiti ništa [3] [5] [6] [7].

Što se tiče nedostataka kod PaaS-a, vezani su za sigurnost podataka jer iako organizacije mogu pokretati svoje aplikacije putem PaaS-a, podatci koji se nalazi na serverima oblaka trećih strana predstavljaju sigurnosne probleme. Integracije su također jedan od problema PaaS-a zbog kompleksnosti povezivanja spremljenih podataka unutar podatkovnog centra koji može biti na stranici ili negdje drugdje. Ovo može uzrokovati problem usvajanja aplikacija i usluga s PaaS ponudom. Nadalje, dolazi do problema jer korisnik ovisi o infrastrukturi i softveru dobavljača pa tako sve promjene ili mogući kvarovi kod dobavljača mogu utjecati na projekte korisnika i ugroziti ih. Na primjer, ako davatelj usluga odluči promijeniti određeni programski jezik i korisnici će biti prisiljeni na to ili će morati početi koristiti jezik drugog davatelja usluga [3] [5] [6] [7].

Primjeri PaaS-a su: AWS Elastic Beanstalk, Google App Engine, Windows Azure, Oracle Cloud Infrastructure, IBM Cloud, Magento Commerce, Quickbase, OutSystems, Boomi, Heroku, Force.com, OpenShift… [3] [5] [6] [7]

5. Usporedba platformi za razvoj aplikacija u oblacima

5.1 AWS Elastic Beanstalk



Slika 4: AWS Elastic Beanstalk Logo, preuzeto s: https://marketplace.microfocus.com/appdelivery/content/deployment-automation-elastic-beanstalk-plugin

AWS Elastic Beanstalk je platforma laka za korištenje te postavljanje web aplikacija i usluga razvijenih uz pomoć programskih jezika Java, .NET, Python, Ruby, Go, PHP, React i Docker na poslužiteljima poput Apache-a, Nginx-a, IIS-a te Passengera. Korisnik jednostavno može prenijeti svoj kod, a Elastic Beanstalk će automatski upravljati implementacijom, sve od osiguravanja kapaciteta, uravnoteženja opterećenja, automatskog skaliranja pa do praćenja stanja same aplikacije. Istovremeno, korisnik zadržava potpunu kontrolu nad AWS resursima koji pokreću aplikaciju te im može pristupiti u bilo kojem trenutku. Za korištenje Elastic Beanstalk-a ne plaća se dodatna naknada već se plaćaju samo AWS resursi potrebni za pohranu i pokretanje aplikacija [12] [13] [14] [15].

Prve od prednosti ove usluge su brzina i jednostavnost korištenja. Upravo je Elastic Beanstalk najbrži i najjednostavniji način na koji se aplikacija može postaviti na AWS. Koristi se AWS Management Console, Git spremište ili integrirano razvojno okruženje (IDE) kao što su Eclipse ili Visual Studio za prijenos same aplikacije, dok Elastic

Beanstalk upravlja implementacijom osiguravanja kapaciteta, uravnoteženjem opterećenja, automatskim skaliranjem te praćenjem zdravlja aplikacija. Na taj način u roku od samo nekoliko minuta aplikacija će biti spremna za korištenje bez vašeg rada na konfiguraciji infrastrukture ili resursa [12] [13] [14] [15].

Osim toga, Elastic Beanstalk omogućava puno veću produktivnost programera. Naime, to je usluga orkestracije koja isključuje neke od hardverskih resursa i pojedinosti kao što je na primjer postavljanje AWS komponenti, a programerima i dalje daje puno izbora vezano za operativni sustav i programski jezik. Na taj način omogućuje im razvijanje i postavljanje aplikacija dok se Beanstalk bavi upravljanjem i konfiguriranjem poslužitelja, baza podataka, uravnotežitelja opterećenja, vatrozida te mreža. Još jedna od prednosti je to što Elastic Beanstalk automatski skalira aplikaciju na temelju njenih potreba uz pomoć lako podesivih postavki. Na primjer, mogu se koristiti mjerni podaci o upotrebi procesora za pokretanje radnji automatskog skaliranja. S ovom platformom, aplikacija može podnijeti i najveće opterećenje ili promet uz smanjenje troškova. Sljedeća prednost je što imamo mogućnost cjelokupne kontrole nad AWS resursima koji pokreću aplikaciju, a uz to ako odlučite preuzeti neke ili sve elemente infrastrukture to lako možete učiniti koristeći Elastic Beanstalk [12] [13] [14] [15].

Što se tiče nedostataka kod Elastic Beanstalk-a prvi je nepouzdana implementacija, odnosno često dolazi do kvarova bez ikakvih naznaka zbog čega, što dovodi do komplikacija u radu te nepotrebnog gubitka sredstava i vremena. Također problem je i spora implementacija koja se ponekad može oduljiti i do petnaest minuta, a s više poslužitelja može potrajati i duže [12] [13] [14] [15].

Nedostatak je i mnoštvo nadogradnji jer Elastic Beanstalk često izlazi u novim verzijama bez da se prethodno obavijesti korisnike o tome što se točno promijenilo. Uz to, dolazi do povremenog brisanja starih verzija aplikacija jer svakim pokretanjem Elastic Beanstalk arhivira staru inačicu aplikacije u S3 kantu, a ako na primjer postoji 500 starih verzija daljnje implementacije ne mogu uspjeti. Ovi svi nedostatci ukazuju na generalni nedostatak transparentnosti kod Elastic Beanstalk-a što je uz sveukupni nedostatak jasne

dokumentacije o nadogradnji mnoštva aplikacija još jedna od mana ove platforme [12] [13] [14] [15].

5.2 Windows Azure



Slika 5: Microsoft Azure Logo, preuzeto s: https://i.pcmag.com/imagery/reviews/04hAlUkn5TfWzY64NOsbIe6-7.1569482813.fit_scale.size_1028x578.jpg

Windows Azure platforma u računalnom oblaku nam nudi više od dvjesto proizvoda i servisa unutar računalnog oblaka dizajniranih kako bismo mogli realizirati svoje ideje. Pruža nam mogućnost da koristimo alate i programske jezike po našem izboru što je veoma pozitivna i fleksibilna stvar [16] [17] [19].

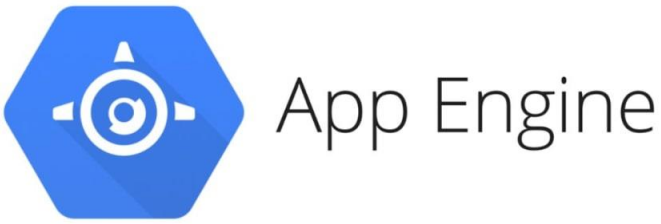
Azure nam pruža garantiranu sigurnost što potkrepljuju višeslojnom sigurnošću koja je postavljena u fizičkim podatkovnim centrima, infrastrukturi i operacijama u Azure-u te nudi zaštitu od DDoS-a [16] [17] [19].

Također, Azure ima mrežu partnera s provjerenim Azure mogućnostima kako bi imali mogućnost birati nekog od njih kako bismo uspjeli lakše doći do naših ciljeva. IT partneri koriste Azure kako bi postavili, upravljali te podržavali potrošačeve već postojeće ideje, a uz to nude i ponude gotovih ideja ili mogućnost izrade prilagođenih ideja, to jest rješenja. Veličina naše tvrtke, kompanije ili sličnog nije bitna jer Azure podržava sve

veličine kako bi se uspjela izvršiti digitalna transformacija. Kao što je i ranije rečeno, Azure podržava tehnologije otvorenog koda kako bismo mogli koristiti alate i tehnologije koje mi želimo što uključuje i programske jezike po našoj želji. Dakle, možemo koristiti, odnosno pokrenuti koju god želimo našu aplikaciju koristeći naš programski kod te naš operacijski sustav na našem uređaju [16] [17] [19].

Što se tiče cijene i plaćanja, plaćamo samo ono što koristimo te nam uz ovo Azure nudi fleksibilnu kupovinu raznih opcija koje zadovoljavaju naše potrebe. Nude se na korištenje i alati koji nam mogu pomoći s potrošnjom u računalnom oblaku. Postoji opcija probnog perioda koje započinje kada kreiramo korisnički račun čime dobivamo pristup besplatnim Azure proizvodima i servisima te uz to dobivamo i dvjesto dolara kredita na korištenje za tih trideset dana probnog perioda. Ukoliko želimo nastaviti koristiti Windows Azure, morati ćemo početi plaćati uslugu u iznosu koji ovisi o tome koliko koristimo naše resurse, odnosno “pay-as-you-go” [16] [17] [19].

5.3 Google App Engine



Slika 6: Google App Engine Logo, preuzeto s: https://hsto.org/webt/4m/mp/dz/4mmpdzaun1lqnfbqlluwok_kiie.jpeg

Google App Engine nam nudi visoku skalabilnost čak do svjetskih razmjera, oslobađa developere od održavanja servera te konfiguriranja aplikacije prilikom postavljanja. Ova platforma kao servis također pruža potporu za popularne programske

jezike i širok spektar developerskih alata. Nudi se pouzdano i brzo korištenje aplikacija i pod velikim teretom te velikom količinom podataka unutar standardnog okruženja [18].

Nadalje, što se tiče programiranja, nudi brzu izgradnju te postavljanje aplikacija napisanog u programskom jeziku po našoj želji te nam uz ovo nudi i upravljanje resursima putem naredbenog retka, “debagiranje” izvornog koda te lako pokretanje API “back-endova”. Neki od programskih jezika koji su podržani su Node.js, Java, Ruby, C#, Go, Python, PHP i ostali. Postoje prilagođena izvođenja što znači da možemo putem Docker kontejnera podignuti bilo koju biblioteku ili “framework”. Također, pomoću “Cloud Monitoring-a” i “Cloud Logging-a” možemo pratiti zdravlje i performanse naše aplikacije,a s “Cloud Debugger-om” te “Error Reporting-om” možemo lako i brzo dijagnosticirati i popraviti greške u kodu. Nudi nam i lako postavljanje različitih verzija naše aplikacije te kreiranje raznih okruženja, kao što su razvoj, test, produkcija te izvedbeno okruženje [18].

Također, možemo koristiti dijeljenje prometa što nam omogućava da dolazeće zahtjeve usmjeravamo na različite verzije aplikacije, da vršimo A/B testove te da vršimo inkrementalna uvođenja značajki. Google App Engine također nudi upute za razne stvari, kao što su osnove App Engine-a, kako u njemu kreirati te postaviti “Hello World” aplikaciju koristeći fleksibilno ili standardno okruženje te programski jezik po našem izboru. Također još nudi upute za dva okruženja za Python programski jezik čime nam pomažu odrediti koji je od ta dva za nas. Jednako tako postoje upute za dva okruženja za Java programski jezik, Node.js programski jezik te postoje upute kako kreirati Android aplikaciju koristeći Firebase i App Engine, koristeći “backend” pohranu podataka, sinkronizaciju u stvarnom vremenu te praćenje korisničkih radnji putem Firebase-a [18].

Sigurnost je ostvarena na način da se od prijetnji štitimo koristeći mogućnosti, odnosno definiranjem pristupnih pravila, App Engine vatrozida, IAM pravila te pravilno korištene i postavljene SSL/TLS certifikate na našoj domeni uz bez dodatnih troškova [18].

Što se tiče plaćanja, plaćaju se samo resursi koje iskoristimo te nam uz ovo sve Google App Engine nudi automatsko skaliranje ovisno o prometu naše aplikacije te koristi resurse samo dok se naš kod koristi. Ukoliko želimo samo isprobati ove funkcionalnosti, Google App Engine nudi na devedeset dana tristo dolara kredita te pristup do preko dvadeset uvijek besplatnih proizvoda. Ako želimo nastaviti koristiti ovu uslugu, morati ćemo ju plaćati po satu po instanci, a cijene variraju ovisno o serveru koji će nam služiti kao poslužitelj. Također, na web stranici Google App Engine-a postoji kalkulator kojim možemo lako izračunati kolika bi plaćali uslugu mjesečno temeljem unesenih podataka [18].

5.4 IBM Cloud



Slika 7: IBM Cloud Logo, preuzeto s: https://developer.ibm.com/recipes/wp-content/uploads/sites/41/2018/03/ibm_cloud-400x209.png

IBM Cloud je “full stack” platforma u računalnom oblaku koja omogućava kompanijama da rade putem javnog, privatnog te tradicionalnog okruženja. Nadalje, nudi mogućnost upravljanja računalnim oblacima drugih dobavljača. Na IBM Cloud-u možemo pronaći više od 170 proizvoda i servisa koji pokrivaju podatke, umjetnu inteligenciju (AI), kontejnere i ostalo [20].

Mogućnosti programiranja su opisane na način da IBM Cloud-ov “Cloud Foundry” sadržava izvedbe za programske jezike kao što su Java, Node.js, PHP, Python, Ruby, ASP.NET, Tomcat, Go te Swift. Također se nudi i korištenje drugih izvedbi, odnosno programskih jezika putem “community build packs-ova” što su zapravo paketi koji dakle

pružaju “framework” te izvedbenu podršku za aplikacije. Izvedbe organiziraju aplikacije koje se izrađuju u obliku procesa bez stanja što omogućava da se ti procesi brzo pokreću i zaustave, ako dođe do nekog prekida, odnosno neuspjeha, repliciraju se ili se dupliciraju u nekim slučajevima. Nadalje, izvedbe, odnosno “runtimes”, povezuju IBM Cloud service s aplikacijama kao krajnjim točkama te time pružaju svakoj instanci aplikacije znanje kako da se nose s podacima i bitnim zahtjevima [20].

Sigurnost IBM Cloud-a je ostvarena kroz niz proizvoda koji IBM Cloud nudi, a neki od njih su IBM Cloud Hyper Protect Crypto Services koji omogućava enkripciju podataka unutar računalnog oblaka koji su zaštićeni unutar posebnog sigurnosnog modula u obliku hardvera, IBM Cloud Hyper Protect Virtual Servers koji nam pruža autoritet nad virtualnim serverima baziranim na Linux operacijskom sustavu za promet koji sadrži osjetljive podatke ili poslovne IP i još puno ostalih proizvoda [20].

Što se tiče plaćanja, imamo mogućnost kreiranja korisničkog računa besplatno čime dobivamo na korištenje preko četrdeset uvijek dostupnih i besplatnih proizvoda bez vremenskog ograničenja. Također dobivamo dvjesto pedeset i šest megabajta memorije u računalnom oblaku na korištenje te dvjesto američkih dolara kredita pomoću kojih možemo isprobati što god želimo unutar IBM Cloud-a unutar trideset dana probnog perioda kojeg dobivamo [20].

6. Usporedba AWS Elastic Beanstalk-a i Windows Azure-a kao platformi za razvoj aplikacija

Odabrani su AWS Elastic Beanstalk i Windows Azure kao dva predstavnika PaaS-a te će na svakog biti učitana jednostavna aplikacija u React programskom jeziku na temelju čega će biti izvršeno testiranje i usporedba ovih servisa. Ova dva predstavnika su bila izabrana jer su se činili najzanimljivijima prilikom istraživanja o njima te njihovog opisivanja prilikom izrade završnog rada.

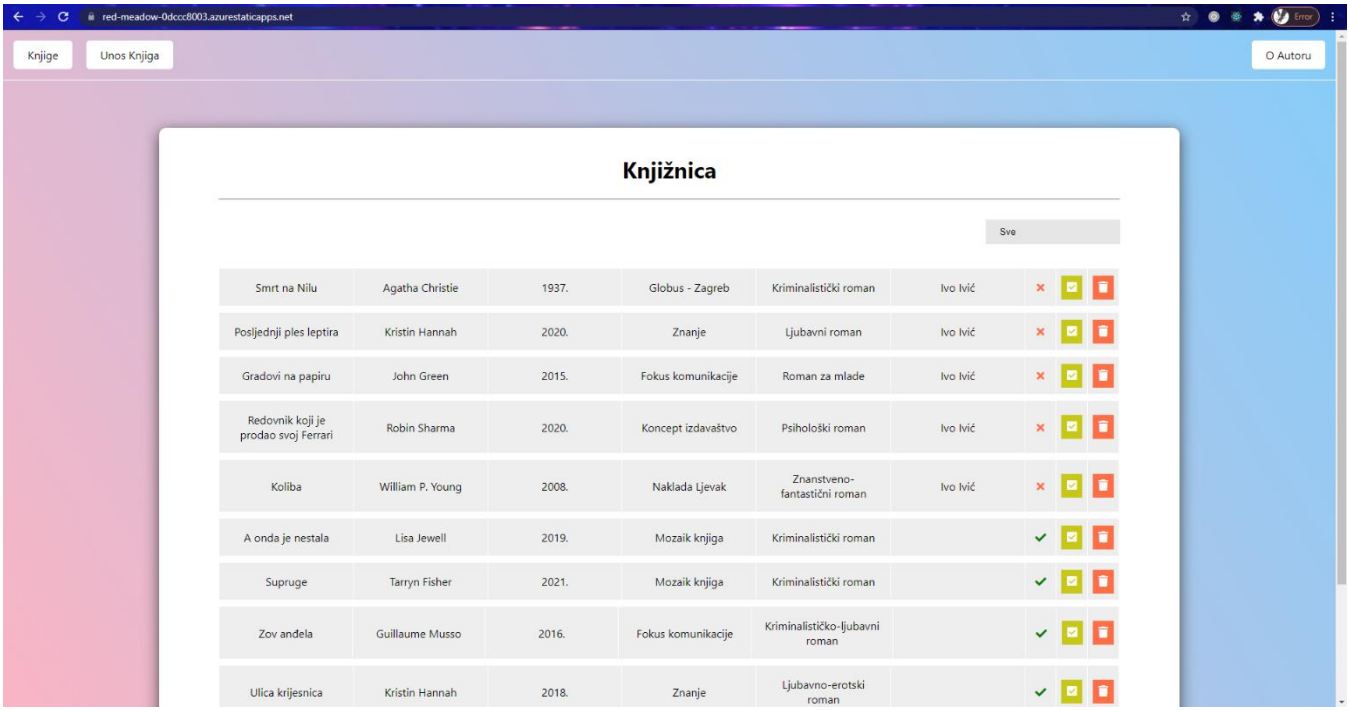
6.1. Microsoft Azure

Što se tiče implementacije aplikacije, Microsoft Azure je bio prvi na redu te je ostavio dobar utisak iz razloga što je bilo lako postaviti web aplikaciju te ju pokrenuti. Prilikom kreiranja korisničkog računa, postupak je bio ispuniti osnovne informacije o korisniku te potvrditi identitet putem koda koji možemo dobiti ili pozivom ili sms porukom koju dobijemo na mobitelu te je nakon toga bilo potrebno unijeti podatke o kartici u svrhu identifikacije korisnika. Naplate nije bilo, te na stranici unosa podataka o kartici piše kako će naplata biti izvršena samo u slučaju da potrošimo sve kredite, koje smo dobili putem besplatnog perioda, putem tarife “pay-as-you-go”, odnosno koliko potrošimo resursa. Postupci su veoma dobro objašnjeni od strane Microsoft-a te nema baš puno mjesta za pogriješiti.

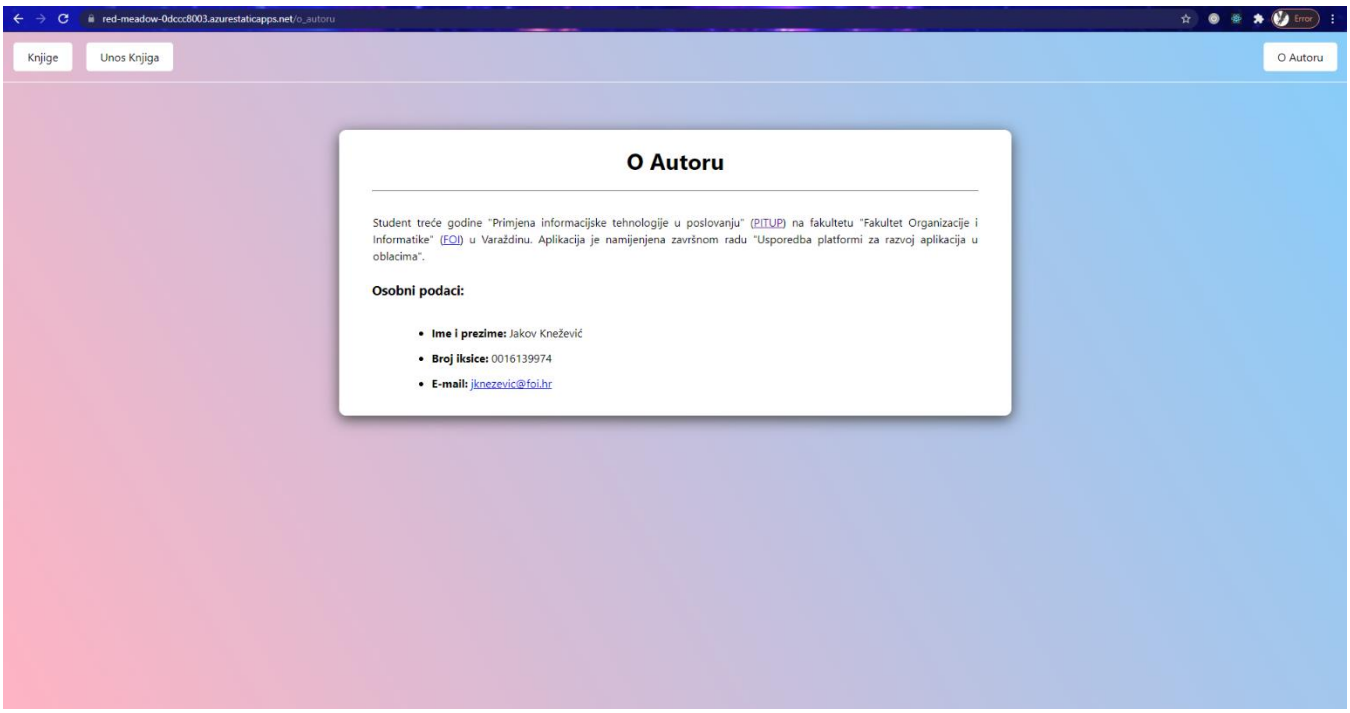


Slika 8: Microsoft Azure Logo 2, preuzeto s: https://digitalample.com/wp-content/uploads/2019/05/cloud-computing-780x405.png

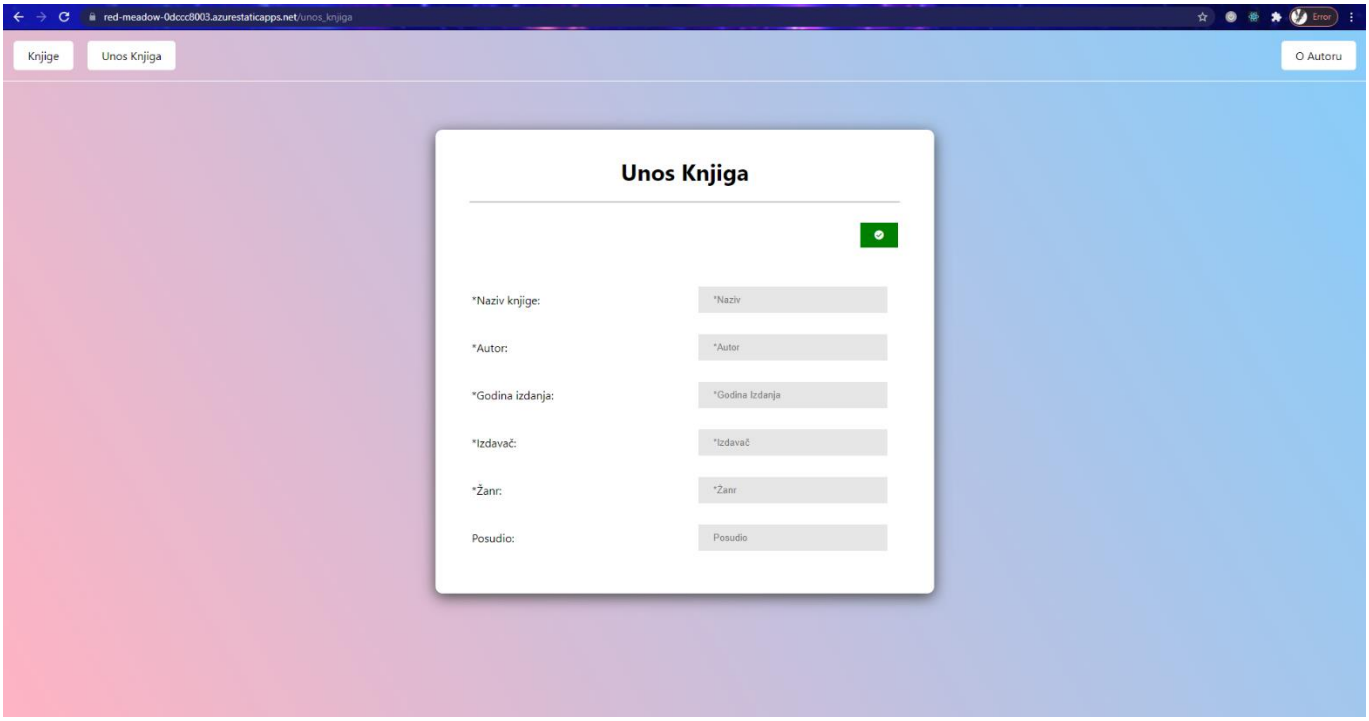
Dakle, kreirana je vrlo jednostavna aplikacija u React programskom jeziku pod nazivom “Knjižnica”. Projekt je bilo potrebno učitati na Github u repozitorij koji smo kasnije prilikom izrade aplikacije u Microsoft Azure-u povezali nakon čega je Microsoft Azure učinio sve potrebno kako bi naša aplikacija proradila. Web aplikacija prikazuje listu knjiga te neke osnovne informacije o njima što nam je prva web stranica aplikacije, a druga web stranica je stranica o autoru gdje su navedene neke osnovne informacije o autoru web aplikacije. Na trećoj stranici imamo unos knjiga gdje možemo unesti novu knjigu u listu. Sve stranice su prikazane na slikama ispod.



Slika 9: Microsoft Azure Knjižnica početna stranica, vlastiti rad



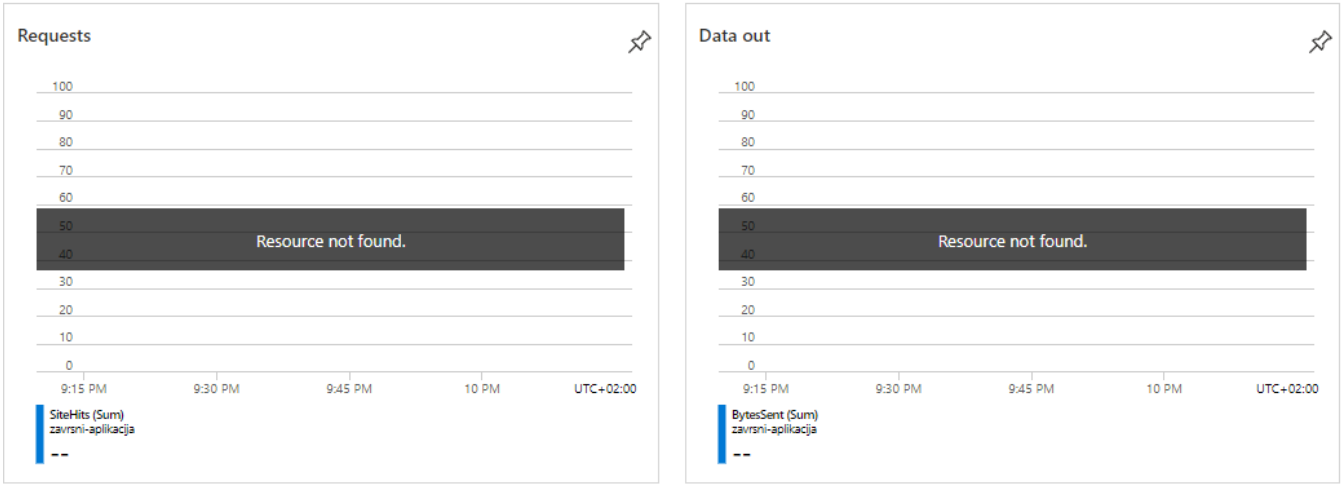
Slika 10: Microsoft Azure Knjižnica stranica o autoru, vlastiti rad



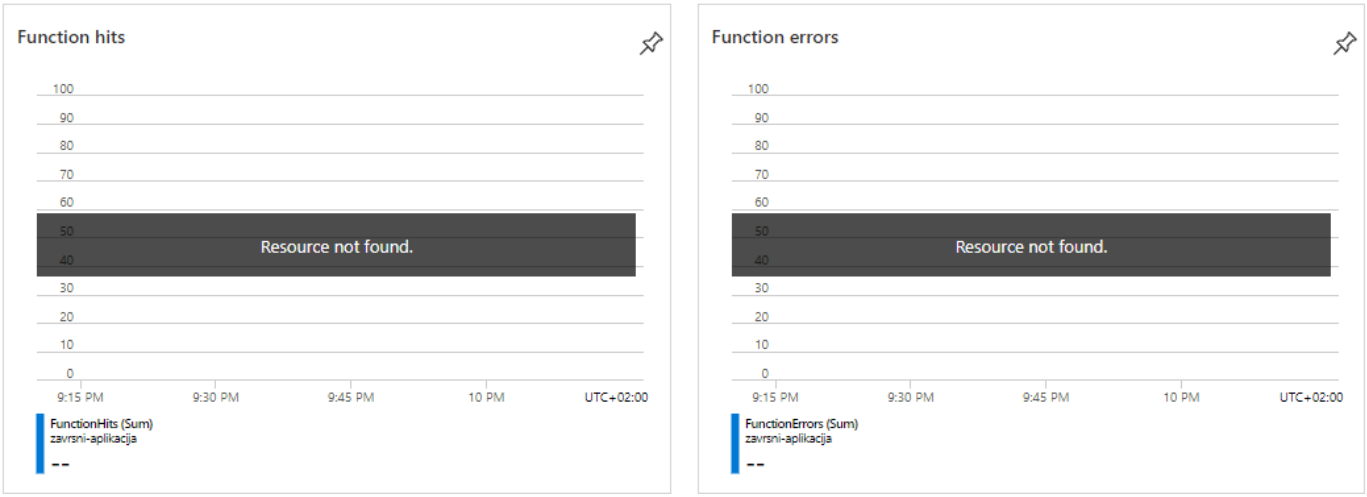
Slika 11: Microsoft Azure Knjižnica stranica za unos knjiga, vlastiti rad

Što se tiče brzine odaziva poslužitelja, ona je poprilično dobra te nije bilo nekih problema s kašnjenjem ili sličnim. Nadalje, Microsoft Azure nam pruža raznu statistiku vezanu uz našu aplikaciju, kao što je statistika koliko puta je netko posjetio našu web stranicu, statistika odlaznih podataka i ostali. Na slikama ispod mogu se vidjeti primjeri raznih vrsta statistika.

“Requests” nam pokazuje koliko puta je našu stranicu netko posjetio, “Data Out” prikazuje statistiku sume poslanih podataka u bajtovima u određenom trenutku. “Function hits” nam prikazuje sumu izvršenih zahtjeva na web stranici u određenom trenutku, a “Function errors” nam prikazuje sumu grešaka s funkcijama na web stranici u određenom trenutku. Na grafovima trenutno nema podataka jer je aplikacija zbog poteškoća morala biti brisana te ponovno postavljana.

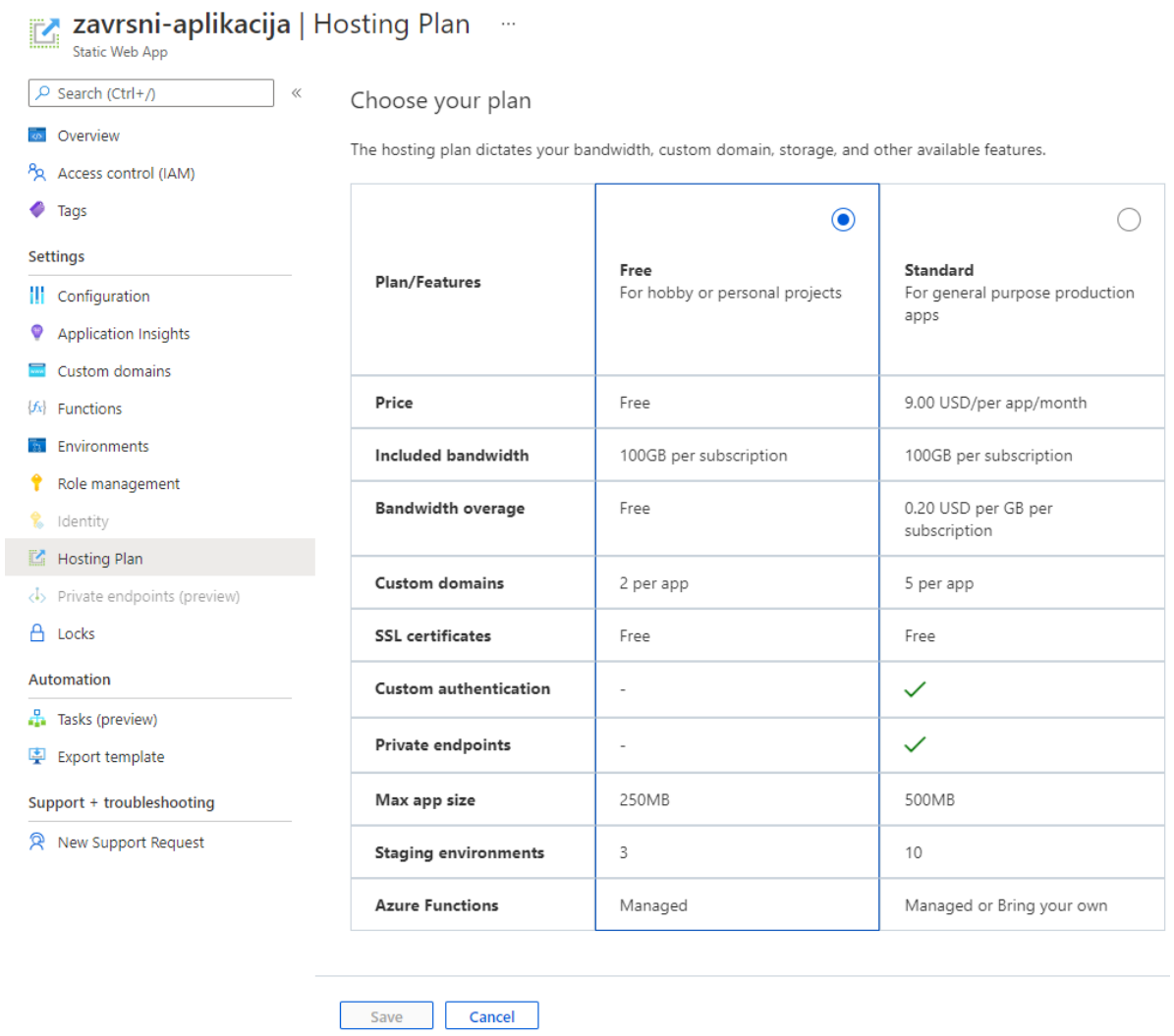


Slika 12: Microsoft Azure Izvještaj 1, preuzeto sa stranice osobnog projekta: https://portal.azure.com/

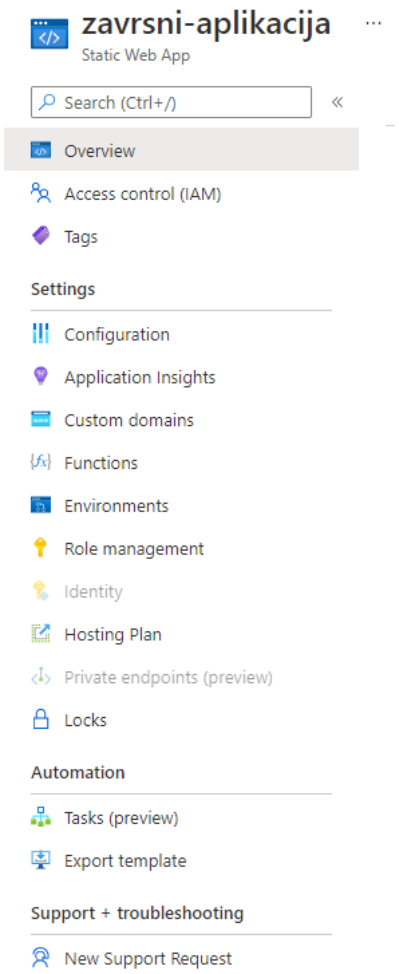


Slika 13: Microsoft Azure Izvještaj 2, preuzeto sa stranice osobnog projekta: https://portal.azure.com/

Na slici ispod možemo vidjeti izbornik unutar naše aplikacije gdje imamo na izbor nekoliko mogućnosti kao što su “Access Control (IAM)” gdje možemo definirati tko može ili ne može pristupiti našoj aplikaciji i slično. Većina mogućnosti su za neke naprednije stvari dok je još jedna od potencijalno korisnih mogućnosti “Hosting Plan” gdje možemo po potrebi mijenjati plan što možemo vidjeti na slici ispod.



Slika 14: Microsoft Azure plan plaćanja, preuzeto sa stranice osobnog projekta: https://portal.azure.com/



Slika 15: Microsoft Azure Izbornik 1, preuzeto sa stranice osobnog projekta: https://portal.azure.com/

6.2. AWS Elastic Beanstalk

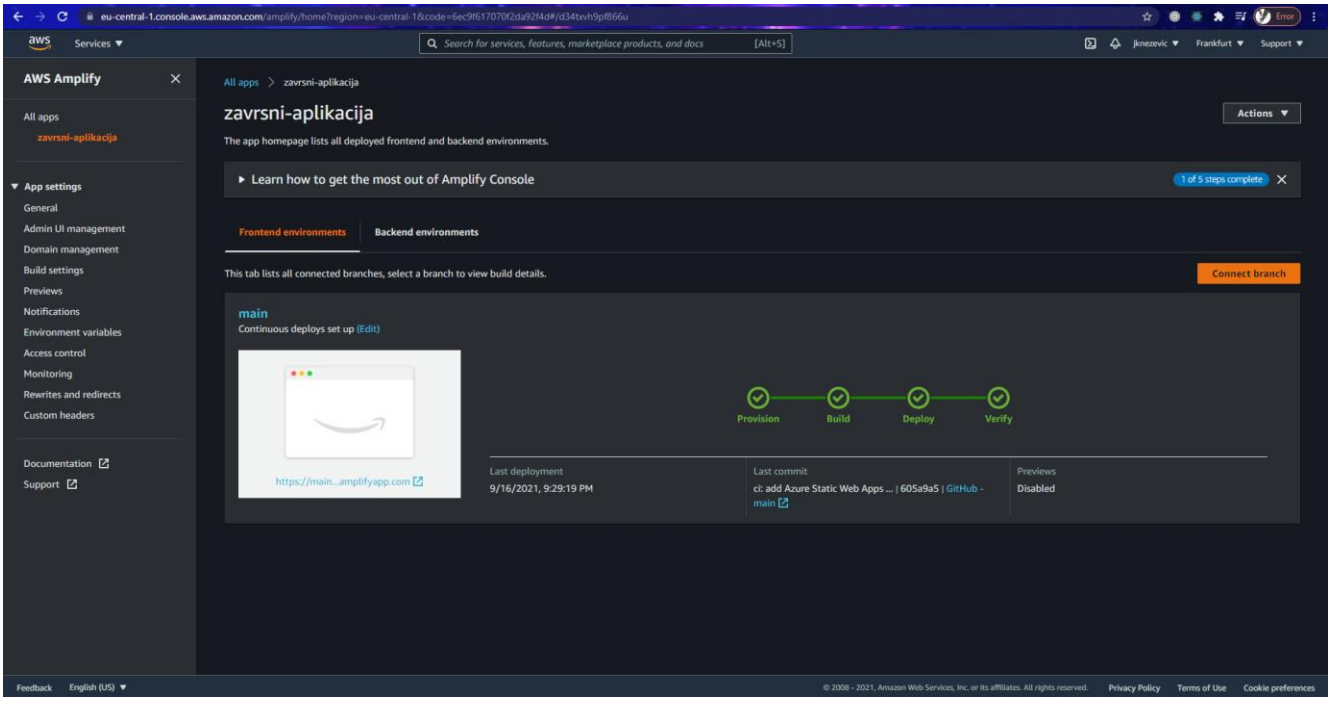
S druge strane, AWS Elastic Beanstalk je možda čak i malo jednostavniji što se tiče implementacije programskog koda na računalni oblak, dakle bilo je potrebno samo naš projekt učitati na Github u repozitorij te ga prilikom kreiranja aplikacije povežemo nakon čega AWS Elastic Beanstalk sam dovrši sve što je potrebno. Prilikom kreiranja korisničkog računa, postupak je bio dosta sličan kao kod Microsoft Azure-a, bilo je potrebno unijeti broj mobitela te ga potvrditi pomoću koda koji nam stigne u obliku sms

poruke te je nakon toga bilo potrebno unijeti podatke o kartici sa koje nam AWS Elastic Beanstalk uzme privremeno jedan dolar s računa, na 3-4 dana, u svrhu identifikacije korisnika. Nadalje, AWS nam naplaćuje samo ukoliko prijeđemo limit koji imamo u paketu "AWS Free Tier" što je zapravo taj probni period.

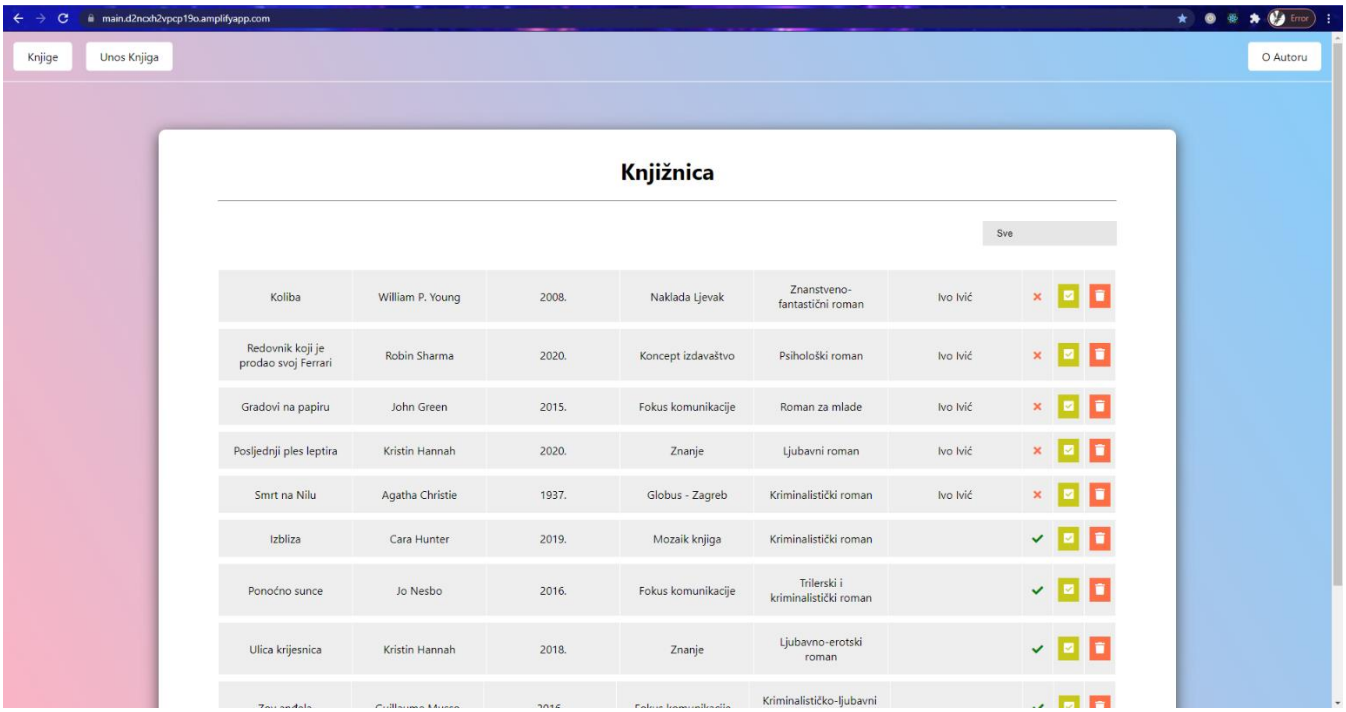


Slika 16: AWS Elastic Beanstalk Logo 2, preuzeto s: https://www.fiverr.com/samdubey/setup-nopcommerce-4-point-1-on-aws-elastic-beanstalk

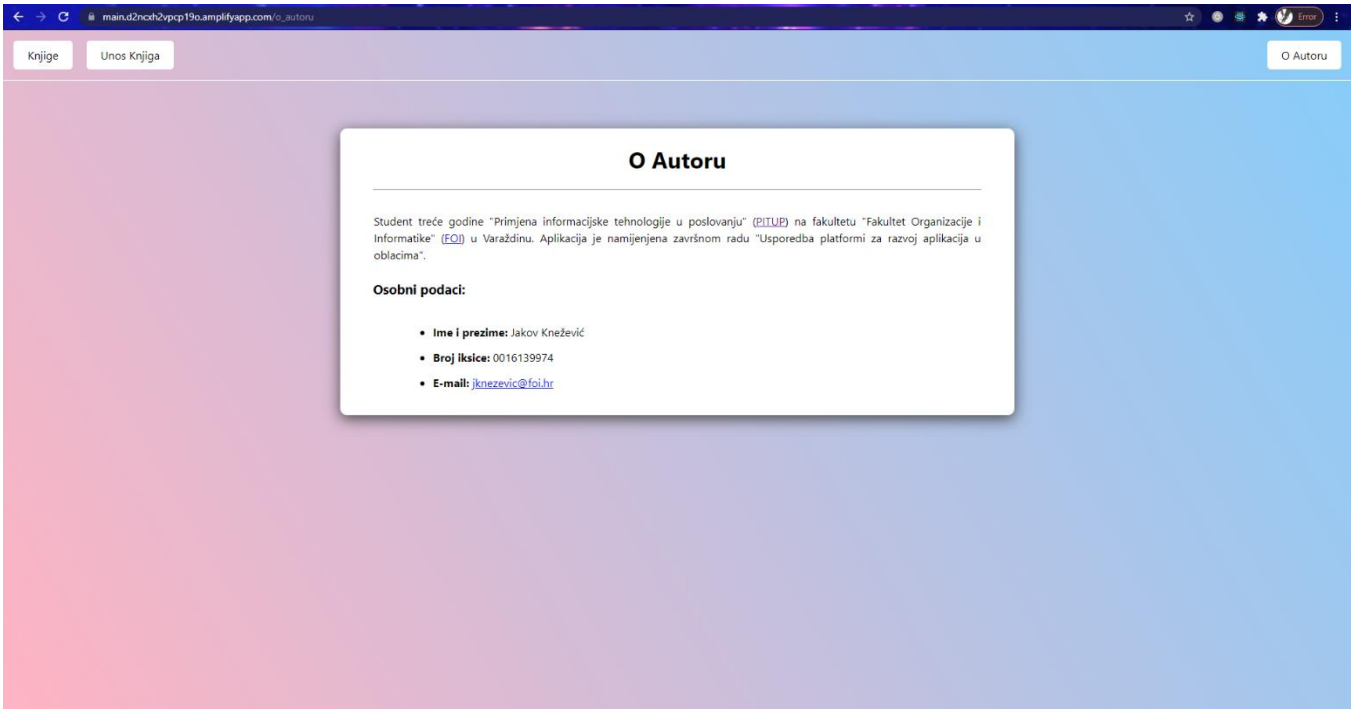
Na Github poslužitelj je u repozitorij učitana identična aplikacija kao i kod Microsoft Azure-a. Time imamo jednu vrlo jednostavnu web aplikaciju napisanu u React programskom jeziku pod nazivom "Knjižnica". Web aplikacija prikazuje listu knjiga te neke osnovne informacije o njima, imamo web stranicu koja sadrži informacije o autoru te također imamo i treću web stranicu koja služi za unos novih knjiga.



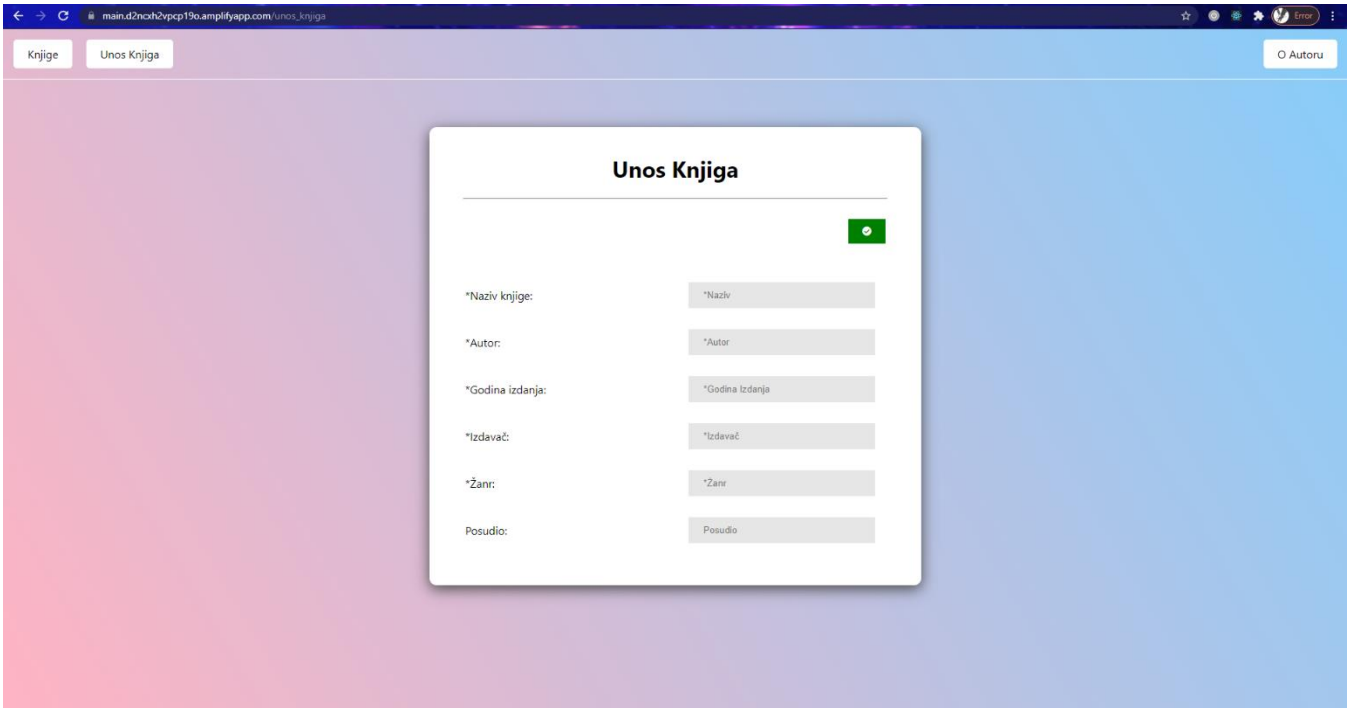
Slika 17: AWS Elastic Beanstalk početna stranica, preuzeto sa stranice osobnog projekta: https://eu-central-1.console.aws.amazon.com/



Slika 18: AWS Elastic Beanstalk Knjižnica početna stranica, vlastiti rad



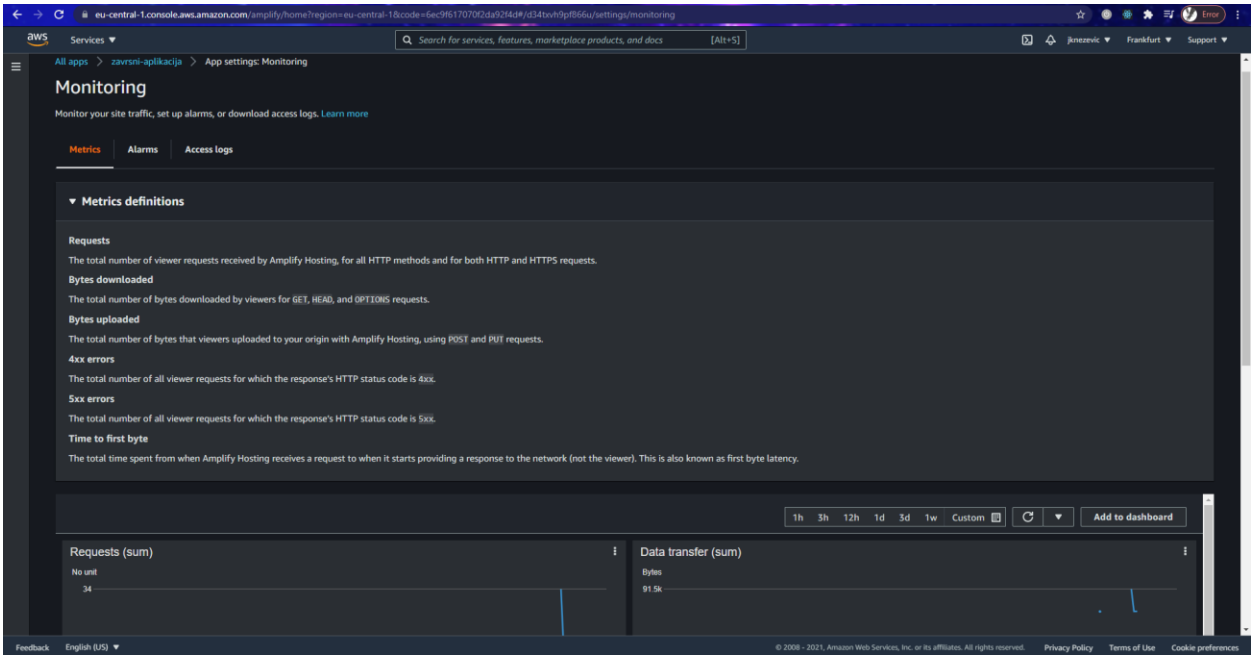
Slika 19:: AWS Elastic Beanstalk Knjižnica stranica o autoru, vlastiti rad



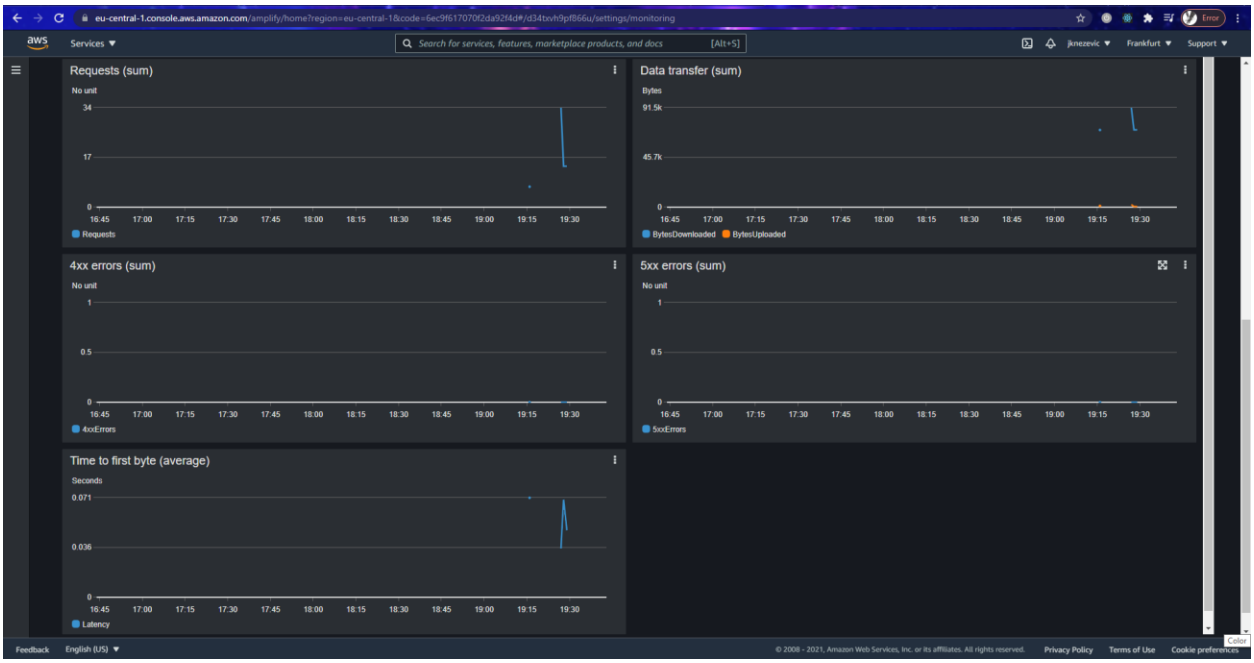
Slika 20: AWS Elastic Beanstalk Knjižnica stranica za unos knjiga, vlastiti rad

Ovdje možemo vidjeti kako je izgled ove web stranice i web stranice u Microsoft Azure-u jednak, kako ne bi bilo zabune ili sumnje kako je ovo ista slika, u snimkama zaslona obje aplikacije se može vidjeti u gornjem lijevom kutu druga adresa web stranice.

Također kao i kod prijašnjeg poslužitelja, nije bilo nikakvih problema oko kašnjenja ili sporog odziva aplikacije. Na slikama ispod možemo vidjeti, jednako kao i kod Microsoft Azure-a, statistiku naše aplikacije.

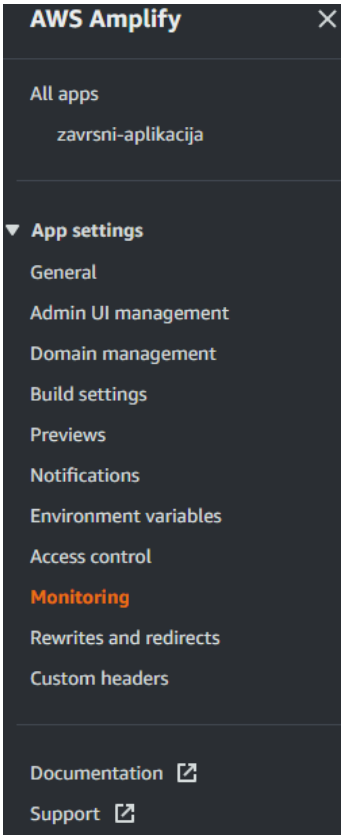


Slika 21: AWS Elastic Beanstalk Izvještaj 1, preuzeto sa stranice osobnog projekta: https://eu-central-1.console.aws.amazon.com/



Slika 22:: AWS Elastic Beanstalk Izvještaj 2, preuzeto sa stranice osobnog projekta: https://eu-central-1.console.aws.amazon.com/

Na prvoj slici možemo vidjeti definicije metrika koje su prikazane na drugoj slici, dakle objašnjeno je što koji graf predstavlja i što on nama znači. Druga slika prikazuje te grafove te možemo vidjeti “Requests (sum)”, “Data transfer (sum)”, “4xx errors (sum)”, “5xx errors (sum)” te “Time to first byte (average)”. “Requests (sum)” nam prikazuje sumu zahtjeva koji su bili izvršeni dok nam “Data transfer (sum)” prikazuje sumu zahtjeva, odnosno podataka koji se šalju ili primaju ovisno o tipu zahtjeva. “4xx errors (sum)” nam prikazuje sumu grešaka pod kodom 4xx što nam znači da prikazuje sumu grešaka pod kodom u razmaku od 400-499, s druge strane imamo “5xx errors (sum)” graf koji je jednak kao i prošli graf samo što razmak nije 400-499, već 500-599. Zadnji graf nam je “Time to first byte (average)” koji prikazuje prosječnu latenciju prvog bajta koji nastane kada AWS primi zahtjev.



Slika 23: AWS Elastic Beanstalk Izbornik 1, preuzeto sa stranice osobnog projekta: https://eu-central-1.console.aws.amazon.com/

Na slici iznad možemo vidjeti ostatak mogućnosti AWS Elastic Beanstalk-a te su nazivi mogućnosti poprilično jednoznačni, dakle ako kliknemo na „Monitoring“ moći ćemo vidjeti sve vezano uz statistiku naše aplikacije koja je bila prethodno opisana. Unutar „Build settings“ možemo vidjeti konfigurirati koje pakete, odnosno programe, naša aplikacija koristi, pod „Access control“ možemo mijenjati tko ima pristup našoj aplikaciji i slično, dok je ostatak opcija za neke naprednije stvari i primjene. Vrlo je jednostavno koristiti i snalaziti se u sučelju AWS Elastic Beanstalk-a što mu je veliki plus.

	AWS Elastic Beanstalk	Microsoft Azure
Jednostavnost korištenja	Veoma je jednostavno koristiti AWS Elastic Beanstalk, izbornik je dobro posložen te nema dvosmislenih podmenija ili sličnoga čime ne gubimo vrijeme na manje bitnim stvarima	Manje jednostavan za korištenje, najviše iz razloga što je teže doći do željenih informacija ili opcija jer se mora proći kroz više menija i podmenija što stvara teži pristup, pogotovo za nove korisnike Microsoft Azure-a. Dakle, u odnosu na AWS Elastic Beanstalk ima sučelje manje prilagođeno korisniku
Jednostavnost postavljanja	Postavljanje je iznimno jednostavno, sve što je potrebno je imati posložen Github repozitorij gdje imamo naš projekt te prilikom kreiranja aplikacije naš Github repozitorij povežemo s AWS-om te nakon toga odaberemo koji programski jezik koristimo, server na kojem će biti aplikacija i slično dok ostatak AWS izvrši sam, a uz to svaki put kada promijenimo neku datoteku ili dodamo novu/e u naš repozitorij, AWS će automatski	Postavljanje i ažuriranje je malo teže i kompliciranije, razlog tome je što kada kreiramo aplikaciju i povežemo ju s našim Github repozitorijem, što je relativno lagano i nema nekih problema s time, moramo pokrenuti “workflow” datoteku na Github stranici koja je zapravo poveznica između Github-a i Azure-a te ona u biti definira koji procesi se moraju izvršiti kako bi se postavila aplikacija. Prvi put se ta datoteka sama izvrši ali

	prepoznati promjene te će ponovno izgraditi aplikaciju s novim datotekama	svaki put kada ažuriramo naš projekt, moramo ažurirane datoteke dodati u novu granu, eng. “branch”, te onda tu granu spojiti, eng. “merge”, sa glavnom granom nakon čega moramo pokrenuti tu “workflow” datoteku. Možda se ne čini kao mnogo posla, no u usporedbi s AWS-om koji se odmah automatski izvrši se čini kako je Azure malo kompliciraniji
Cijena korištenja usluga	AWS Elastic Beanstalk nam nudi besplatne usluge koje su oni nazvali “Free tier” gdje imamo zapravo solidan broj mogućnosti i usluga koje možemo koristiti bez da moramo platiti nešto što je definitivno veliki plus, pogotovo ako se izrađuju ili postavljaju aplikacije u edukacijske, eksperimentalne ili slične svrhe	Microsoft Azure nam u sklopu programa “Free trial” daje na korištenje 200 eura kredita, kako bismo mogli isprobati što god želimo te još jedan plus koji ovime imamo je što time možemo, a ne moramo, koristiti besplatne usluge za procesorsku moć, skladišnu memoriju ili slično jer pomoću ovih kredita možemo isprobati neke usluge i mogućnosti koje se inače naplaćuju

Tablica 1: Usporedba AWS Elastic Beanstalk-a i Microsoft Azure-a

7. Zaključak

Ovaj završni rad nam je pružio dobar uvid u podatkovne centre te kako oni zapravo funkcioniraju te koje odgovornosti oni nose. Kod klasičnih podatkovnih centara smo saznali kako zapravo postoji puno velikih troškova kako bi se izgradio, održavao i ažurirao sam podatkovni centar te nam je za njega potreban visok kapital. S druge strane, saznali smo što su podatkovni centri u računalnim oblacima te što nam oni nude. Također smo saznali kako većina poznatijih podatkovnih centara u oblaku nudi besplatne probne periode kojima možemo dugo i kvalitetno isprobati i koristiti njihove usluge.

Što se tiče samo PaaS-a koji je bio najviše proučavan, saznali smo koji sve poslužitelji postoje te smo saznali što oni nude, koje mogućnosti imaju te kakvu nam kvalitetu usluge nude. Svaka od tih platformi ima svoje prednosti i mane, no jednako tako, svatko traži uslugu po njegovim ukusima i potrebama što znači da nijedna prednost i/ili mana ne čini neku platformu lošijom ili boljom, već su sve one samo različite mogućnosti.

Nadalje, što se same usporedbe tiče, Microsoft Azure i AWS Elastic Beanstalk su ugodno iznenadili jer su veoma laki za korištenje te nije potrebno neko ogromno predznanje ili da nam treba puno vremena za pohvatati sve znanje potrebno. Srećom, za to su zaslužni i sami developeri tih tvrtki koji su osmislili kvalitetne priručnike kojima jednostavno možemo podignuti našu prvu aplikaciju na njihovim serverima.

Microsoft Azure ima oku veoma atraktivno sučelje te se lako dolazi do potrebnih informacija, dosta je pregledno sve i nije se teško snaći iako se možda čini relativno natrpanim na prvi pogled. Njegove mogućnosti također ne posustaju te su na visokoj razini kompetitivnosti s drugim platformama.

AWS Elastic Beanstalk je također veoma lagan za korištenje, lako se dolazi do informacija i uputa te je postavljanje aplikacije na poslužitelj iznimno lako, u par klikova je naša aplikacija spremna za korištenje.

Na kraju svega, obje platforme su lake za korištenje, nude velik izbor mogućnosti te definitivno imaju velik potencijal što može potkrijepiti njihova poznatost i proširenost po svijetu. Podatkovni centri u računalnim oblacima su zasigurno budućnost te uvelike olakšavaju posao te kontroliranje troškova, kako velikih kompanija tako i manjih poduzeća i običnih korisnika koji žele napraviti neku svoju aplikaciju i omogućiti da ju svatko može vidjeti bez da u to mora ulagati ogromne svote novca. Baš radi ovoga je ovaj tip podatkovnog centra u velikoj prednosti nad klasičnim te se zbog toga sve veći broj kompanija, poduzeća i ostali okreću prema računalnim oblacima te podatkovnim centrima u računalnim oblacima.

8. Popis literature

[1] *"What is SaaS?"* [bez dat.] u Salesforce, [Na Internetu], Dostupno: https://www.salesforce.com/in/saas/ [pristupano 05.08.2021.]

[2] "*What is SaaS?"* [bez dat.] u Microsoft, [Na Internetu], Dostupno: https://azure.microsoft.com/en-us/overview/what-is-saas/ [pristupano 05.08.2021.]

[3] S. Watts, M. Raza [15.06.2019.] "*SaaS vs PaaS vs IaaS: What's The Difference & How To Choose"*, [Na Internetu], Dostupno: https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/ [pristupano 05.08.2021.]

[4] T. Short [13.02.2020.] "*What is SaaS? 10 FAQs About Software as a Service"*, [Na Internetu], Dostupno: https://www.softwareadvice.com/resources/saas-10-faqs-software-service/ [pristupano 06.08.2021.]

[5] M. Smith [10.02.2021.] "*What is Platform as a Service? PaaS examples, applications, and advice"*, [Na Internetu], Dostupno: https://www.zendesk.com/blog/what-is-paas/ [pristupano 06.08.2021.]

[6] N. Sakovich [30.05.2019.] "*Advantages and Disadvantages of PaaS: a Practical Overview*", [Na Internetu], Dostupno: https://www.sam-solutions.com/blog/advantages-and-disadvantages-of-paas-practical-overview/ [pristupano 06.08.2021.]

[7] J. Hoffman [bez dat.] "*Advantages and Disadvantages of PaaS You Need to Know*", [Na Internetu], Dostupno: https://wisdomplexus.com/blogs/advantages-disadvantages-paas/ [pristupano 06.08.2021.]

[8] S. Ranger [13.12.2018.] "*Cloud*", [Na Internetu], Dostupno: What is cloud computing? Everything you need to know about the cloud explained | ZDNet [pristupano 06.08.2021.]

[9] J. Sanders, C. Forrest [01.07.2014.] "*The Art of the Hybrid Cloud*", [Na Internetu], Dostupno: Hybrid cloud: What it is, why it matters | ZDNet [pristupano 06.08.2021.]

[10] "*What is a hybrid cloud?"* [bez dat.] u Microsoft, [Na Internetu], Dostupno: What is Hybrid Cloud Computing – Definition | Microsoft Azure [pristupano 09.08.2021.]

[11] "*Data Centre"* [bez dat.] u Sify Technologies, [Na Internetu], Dostupno: How Cloud Datacenters Differ from Traditional Datacenters [pristupano 09.08.2021.]

[12] "*What does Elastic Beanstalk Do and Why use it?*" [bez dat.] u DevelopIntelligence, [Na Internetu], Dostupno: https://www.developintelligence.com/blog/2017/04/aws-elastic-beanstalk-use/ [pristupano 11.08.2021.]

[13] "AWS Elastic Beanstalk" [bez dat.] u AWS Amazon, [Na Internetu], Dostupno: https://aws.amazon.com/elasticbeanstalk/ [pristupano 11.08.2021.]

[14] "*AWS Elastic Beanstalk Reviews*" [bez dat.] u TrustRadius, [Na Internetu], Dostupno: https://www.trustradius.com/products/aws-elastic-beanstalk/reviews?qs=pros-and-cons [pristupano 11.08.2021.]

[15] A. Camp [19.02.2016.] "*Elastic Beanstalk: Advantages and Drawbacks*", [Na Internetu], Dostupno: https://medium.com/@acamp/elastic-beanstalk-advantages-and-drawbacks-be814615af01 [pristupano 11.08.2021.]

[16] "*What is Azure?"* [bez dat.] u Microsoft, [Na Internetu], Dostupno: What is Azure—Microsoft Cloud Services | Microsoft Azure [12.08.2021.]

[17] "*Azure vs. AWS"* [bez dat.] u Microsoft, [Na Internetu], Dostupno: Why Azure vs. AWS | Microsoft Azure [pristupano 12.08.2021.]

[18] "*App Engine*" [bez dat.] u Google, [Na Internetu], Dostupno: App Engine Application Platform | Google Cloud [pristupano 17.08.2021.]

[19] "*App Service documentation"* [bez dat.] u Microsoft, [Na Internetu], Dostupno: https://docs.microsoft.com/en-us/azure/app-service/ [pristupano 17.08.2021.]

[20] "*IBM Cloud*" [bez dat.] u IBM, [Na Internetu], Dostupno: https://www.ibm.com/cloud [pristupano 17.08.2021.]

9. Popis slika

Slika 1: Klasični podatkovni centar vs. podatkovni centar u oblaku 8
Slika 2: Podjela odgovornosti korisnika različitih tipova podatkovnih centara 12
Slika 3: Piramida tipova računalstva u oblacima 19
Slika 4: AWS Elastic Beanstalk Logo 25
Slika 5: Microsoft Azure Logo 27
Slika 6: Google App Engine Logo 28
Slika 7: IBM Cloud Logo 30
Slika 8: Microsoft Azure Logo 2 33
Slika 9: Microsoft Azure Knjižnica početna stranica 34
Slika 10: Microsoft Azure Knjižnica stranica o autoru 34
Slika 11: Microsoft Azure Knjižnica stranica za unos knjiga 35
Slika 12: Microsoft Azure Izvještaj 1 36
Slika 13: Microsoft Azure Izvještaj 2 36
Slika 14: Microsoft Azure plan plaćanja 37
Slika 15: Microsoft Azure Izbornik 1 38
Slika 16: AWS Elastic Beanstalk Logo 2 39
Slika 17: AWS Elastic Beanstalk početna stranica 40
Slika 18: AWS Elastic Beanstalk Knjižnica početna stranica 40
Slika 19:: AWS Elastic Beanstalk Knjižnica stranica o autoru 41
Slika 20: AWS Elastic Beanstalk Knjižnica stranica za unos knjiga 41
Slika 21: AWS Elastic Beanstalk Izvještaj 1 42
Slika 22:: AWS Elastic Beanstalk Izvještaj 2 43
Slika 23: AWS Elastic Beanstalk Izbornik 1 44

10. Popis tablica

Tablica 1: Usporedba AWS Elastic Beanstalk-a i Microsoft Azure-a 46