



Univerzitet u Nišu
Elektronski Fakultet
Katedra za računarstvo



Seminarski rad

Cloud Databases / DBaaS PostgreSQL

Cloud baze podataka i Database as Service rešenja – PostgreSQL (ElephantSQL)
Srđan Ognjanović | DBMS

Sistemi za upravljanje bazama podataka

Profesori: Leonid Stoimenov, Aleksandar Stanimirović

Sadržaj

Sadržaj.....	1
1. Opis.....	2
2. Teorijski deo – <i>Cloud</i> baze podataka.....	3
2.1 Uvod.....	3
2.2 Tipovi <i>cloud computing</i> -a.....	4
2.3 Razlike između tradicionalnih i <i>cloud</i> baza podataka.....	5
2.4 Načini reprezentacije <i>cloud</i> baza podataka.....	6
2.5 Tipovi <i>cloud</i> baza podataka.....	7
2.6 Database as Service (DBaaS).....	10
2.7 Prednosti i nedostaci <i>cloud</i> baza podataka.....	12
3. Praktični deo – ElephantSQL.....	14
3.1 Uopšteno o tehnologiji.....	14
3.2 Planovi baze podataka, cene i servisi.....	15
3.3 Kreiranje instance i navigacioni meni.....	18
3.4 Interakcija sa bazom.....	19
3.5 Mogućnosti i alati.....	21
4. Zaključak.....	24
5. Literatura.....	25

1. Opis

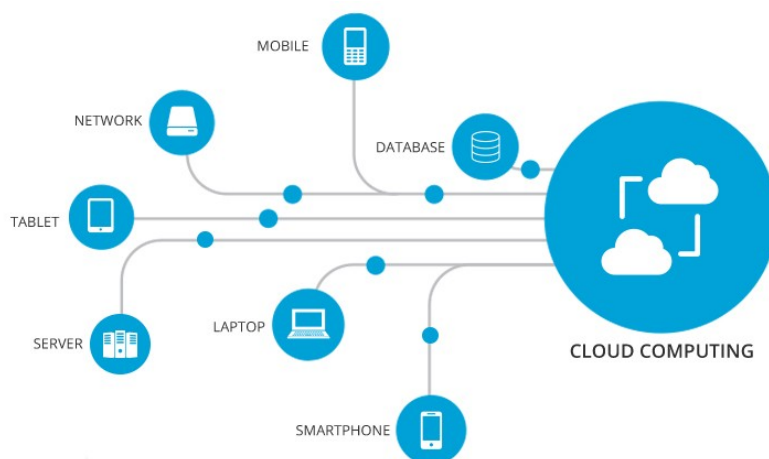
U seminarskom radu biće teorijski obrađene izabrane teme sa opisom praktične implementacije na primeru tehnologije ElephantSQL.

Seminarski rad na temu *Cloud baze podataka i Database as Service rešenja na primeru PostgreSQL-a kao servisa ElephantSQL-a* iz predmeta *Sistemi za upravljanje bazama podataka (DBMS)* se sastoji iz dve celine. Prvi deo predstavlja teorijsku podlogu i ključne koncepte na kojima se zasnivaju računarstvo u oblaku, *Cloud* baze podataka i *Database as service* rešenja. Drugi deo se odnosi na opis i praktičnu realizaciju odnosno primenu PostgreSQL-a u vidu servisa, koji se naziva ElephantSQL. Za potrebe rada podignuta je instanca baze podataka ElephantSQL u vidu *cloud* servisa.

2. Teorijski deo – *Cloud* baze podataka

2.1 Uvod

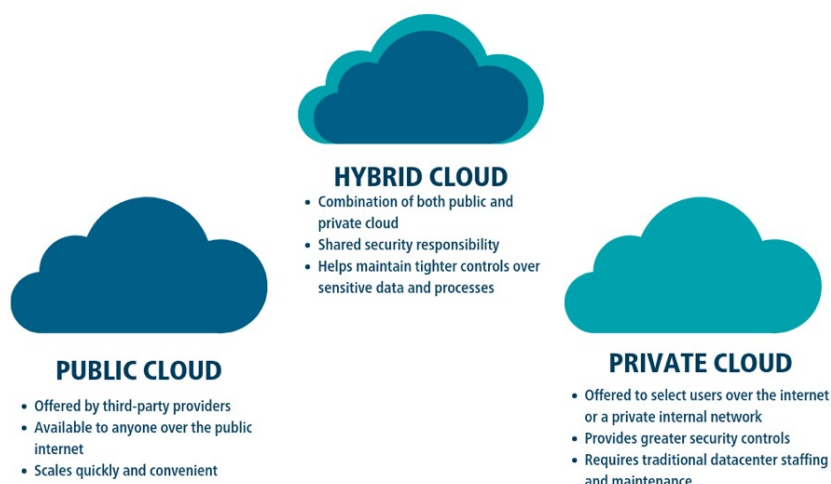
Računarstvo u oblaku (*Cloud Computing*) je paradigma koja obezbeđuje računarsku infrastrukturu platformi i softvera kao usluga putem Interneta. Nudi značajne ekonomske prednosti ograničavanjem kako početnih kapitalnih ulaganja na računarsku infrastrukturu, tako i ukupnih troškova vlasništva. Računarstvo u oblaku uvelo je novi izazov upravljanja petabajtima podataka na skalabilan način. Tradicionalni sistemi baza podataka za upravljanje podacima kompanija pokazali su se neadekvatnim za rešavanje ovog izazova, što je rezultovalo velikom revizijom arhitekture. Predviđa se da će buduća istraživanja za računarstvo u oblaku rezultovati pojavom novih arhitektura upravljanja podacima i međusobnim delovanjem strukturiranih i nestrukturiranih podataka, kao i drugim razvojem [1]. Pojavom ove tehnologije, troškovi računanja, hosting aplikacija, skladištenje sadržaja i isporuka se značajno smanjuju. Računarstvo u oblaku je praktičan pristup iskustvu direktnih troškova i ima potencijal da transformiše centar podataka iz kapitalno intenzivnih postavki u okruženje promenljivih cena [2].



Korišćenjem *cloud*-a kompanije mogu izbeći trošenje sredstava na hardversku infrastrukturu i njeno održavanje, plaćanjem samo određenih usluga prema neophodnim zahtevima i potrebi. Ovim putem se omogućuje obezbeđivanje efikasne skalabilnosti, jer se prema potrebi lako može pribaviti neophodna količina resursa, kao što su *bandwidth*, *memory storage*, *CPU*, itd. Druga pogodnost koju obezbeđuje *cloud* jeste povećanje produktivnosti iz razloga što otklanja potrebu za raznim aktivnostima koje su vezane za održavanje sistema, kao što su postavka hardvera, *software patching*, itd, tako da se IT tim može orijentisati ka važnijim poslovima [3].

2.2 Tipovi *cloud computing*-a

Nisu sve vrste *cloud*-a identične i nije svaki tip računarstva u oblaku za svaku namenu. Nekoliko različitih modela, tipova i usluga evoluiralo je kako bi pomoglo kompanijama da ponude pravo rešenje za njihove potrebe. Najpre je potrebno odrediti vrstu *cloud deployment*-a ili arhitekturu računarstva u oblaku na kojoj će se primenjivati usluge. Postoje tri različita načina primene usluga u oblaku: na javnom, privatnom ili hibridnom oblaku [3].



2.2.1 *Public cloud*

Javni oblaci su u vlasništvu i njima upravljaju nezavisni dobavljači (*third-party*) usluga u oblaku koji svoje računarske resurse, poput servera i skladišta, isporučuju putem Interneta. *Microsoft Azure* je primer javnog oblaka. Sa javnim oblakom, sav hardver, softver i druga prateća infrastruktura je u vlasništvu i njime upravlja dobavljač oblaka. Ovim uslugama se pristupa i upravlja nalogom pomoću veb pretraživača [3].

2.2.2 *Private cloud*

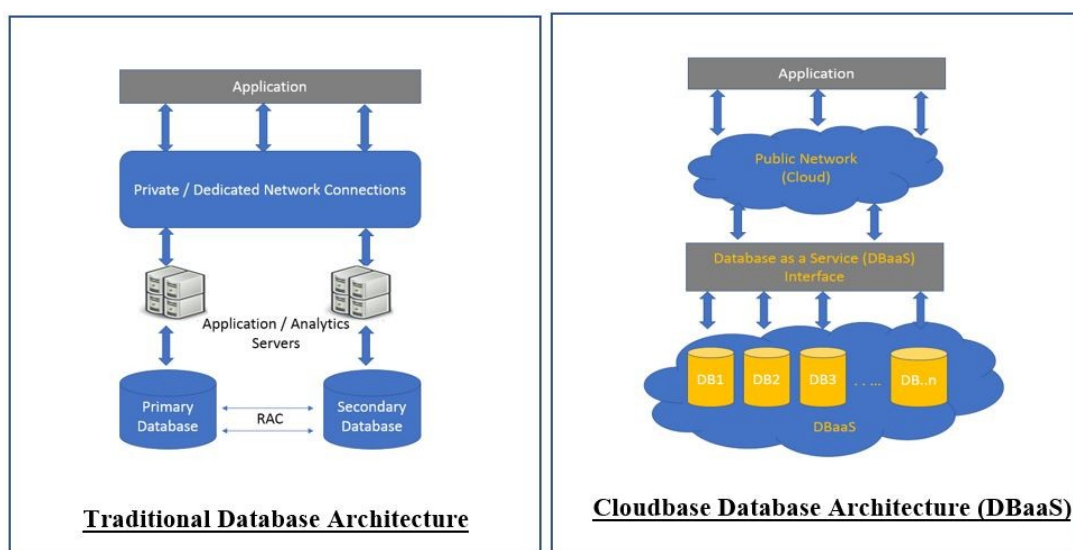
Privatni oblak odnosi se na resurse računarstva u oblaku koje koristi isključivo jedno preduzeće ili organizacija. Privatni oblak može se fizički nalaziti u kompanijskom centru podataka na licu mesta. Neke kompanije takođe plaćaju nezavisne dobavljače usluga da hostuju svoj privatni oblak. Obično je privatni oblak onaj u kojem se usluge i infrastruktura održavaju na privatnoj mreži [3].

2.2.3 *Hybrid cloud*

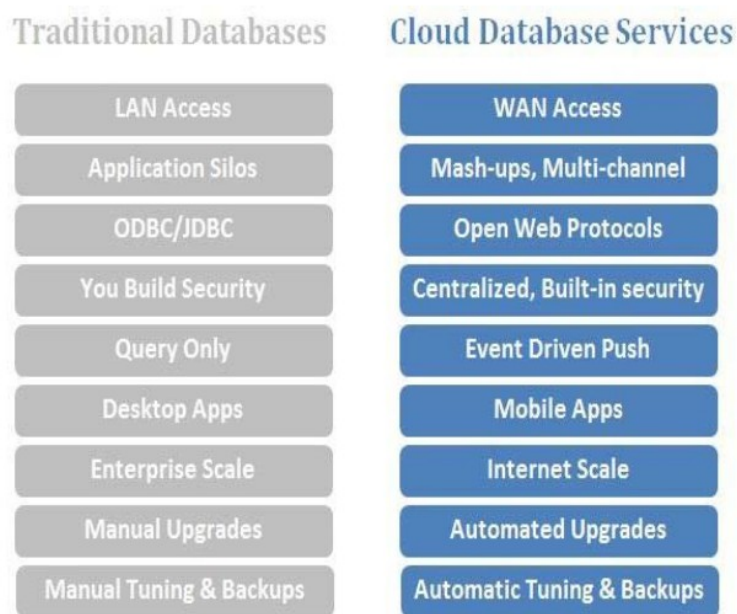
Hibridni oblaci kombinuju javni i privatni oblak, povezani tehnologijom koja omogućava međusobno deljenje podataka i aplikacija. Omogućavajući premeštanje podataka i aplikacija između privatnih i javnih oblaka, hibridni oblak pruža vašem preduzeću veću fleksibilnost, više mogućnosti primene i pomaže u optimizaciji postojeće infrastrukture, sigurnosti i usklađenosti [3].

2.3 Razlike između tradicionalnih i *cloud* baza podataka

Kod tradicionalnih baza podataka, server baze podataka je deo lokalne računarske infrastrukture i za pokretanje, konfigurisanje i održavanje je zadužen IT sektor odnosno korisnik/klijent. Za razliku od ovog modela, kod *cloud* baza podataka, klijent koristi bazu podataka kao uslugu (servis), koja je obezbeđena od strane *cloud* provajdera, a ne kao produkt, što znači da instalacija, inicijalno konfigurisanje i održavanje baze ne spada u domen odgovornosti korisnika, već onog ko pruža uslugu, u ovom slučaju pomenutog *cloud* provajdera.



Na slikama iznad dat je prikaz standardne i *cloud* arhitekture baze. Na levoj slici aplikacioni i serveri baze su u vlasništvu kompanije, kao i konekcija između aplikacije i baze. Na slici desno aplikacija putem javne mreže odnosno *cloud*-a, preko odgovarajućeg API-a, komunicira sa bazom podataka. Konfigurisanje instanci baza koje su na *cloud*-u se obavlja uz pomoć web konzole. Što se skalabilnosti tiče, različiti provajderi koriste različita rešenja. Neki koriste svojstvo *autoscale* kojim se skalabilnost automatski prilagođava količini saobraćaja, dok kod drugih *cloud* baza, korisnik putem API-a može da utiče na skalabilnost [5].



Na slici iznad su ilustrovane neke od razlika između tradicionalnih i *cloud* baza podataka [2].

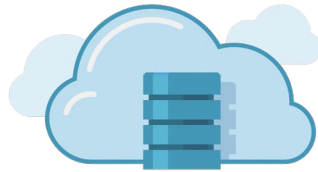
Na slici ispod kroz tabelarni prikaz su dati pregled i poređenje tradicionalnih i *cloud* baza podataka [5].

Measures	On-Premises Database	Cloud Database / DBaaS
Reliability	Reliable, and Private.	More reliable but not necessarily Private.
Scalability	Limited scalable.	Unlimitedly scalable.
Speed	Faster, but may fail in any point of time (in care of hardware failure).	Faster and will be up always.
Deployment	Deployment takes time.	Deployed within no time.
Cost Effectiveness	Lots of capital required to setup on-premises database as a service.	Pay only for what you Use. Highly Cost effective. No Overhead cost involved.
Maintenance	High on Maintenance Cost. All cost to the company. HW, technicians, DBA's and other infrastructure.	No Maintenance Cost. Pay for what you use.
Setup Cost	Entire Setup cost is to be borne by the Company.	Entire Setup cost is borne by Vendor. The company pays only for the Service.
Security	Highly Secure and Controlled.	Highly Secured as per Vendor.

2.4 Načini reprezentacije *cloud* baza podataka

Postoje dva načina za pokretanje baze podataka na cloud-u [4]:

- Preko virtuelne mašine (VM)
- Putem servisa (DBaaS)



2.4.1 Pokretanje preko virtuelne mašine

Pokretanje preko virtuelne mašine se vrši na taj način što korisnik zakupljuje virtuelnu mašinu od provajdera na određeni vremenski period, na kojoj pokreće server baze. Kod ove vrste implementacije, korisnici koriste svoj *machine image* na kome instaliraju bazu, ili gotov *machine image* sa već instaliranom optimalnom bazom. Na primer Oracle obezbeđuje gotov *machine image* sa instaliranim Oracle Database 11g Enterprise Edition bazom koja se pokreće na Amazon EC2 *cloud* platformi. Kod ovakvog pristupa, provajder praktično obezbeđuje server (hardver), dok je korisnik zadužen za održavanje baze podataka [4, 5].

2.4.2 Pokretanje putem servisa

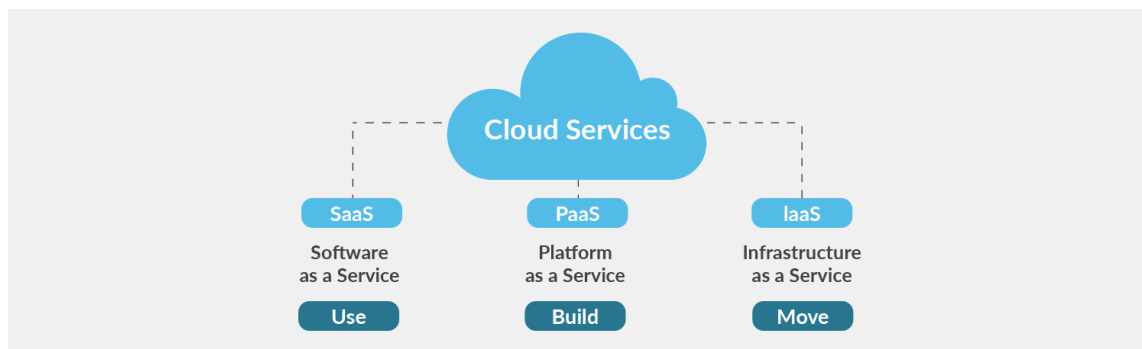
Drugi način implementacije *cloud* baze jeste putem servisa (DBaaS). Kod ovog pristupa korisnik dobija bazu u obliku servisa, što znači da korisnik ne mora da pokreće virtuelnu mašinu i da na njoj instalira i održava bazu. Sve ovo, za korisnika obezbeđuje provajder. Kod ovakvog pristupa korisnik plaća servis na osnovu toga koliko je koristio bazu. Primeri ovakve implementacije bili bi: Google's BigTable, Amazon Relational Database Service i Azure SQL Database [5]. Ovaj način reprezentacije *cloud* baze biće detaljnije opisan u nastavku.

2.5 Tipovi *cloud* baza podataka

Većina usluga računarstva u oblaku spada u četiri široke kategorije:

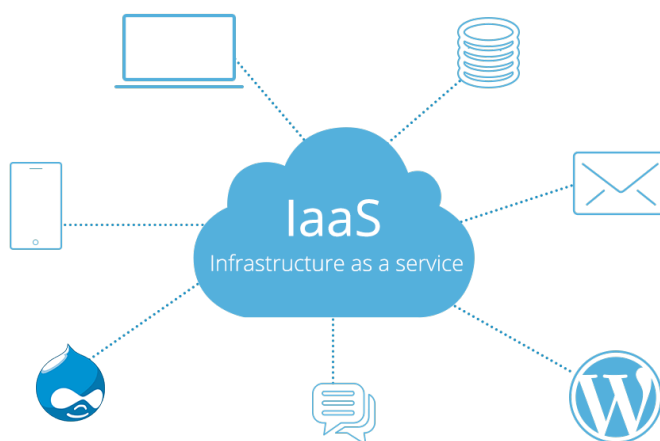
- Infrastruktura kao usluga (*IaaS*)
- Platforma kao usluga (*PaaS*)
- Bez servera (*Serverless*)
- Softver kao usluga (*SaaS*)

Oni se ponekad nazivaju *stack*-om računarstva u oblaku, jer se grade jedan nad drugim. Korisnik bi trebalo da pozna je šta su i kako se razlikuju, kako bi lakše ostvario svoje poslovne ciljeve [3].



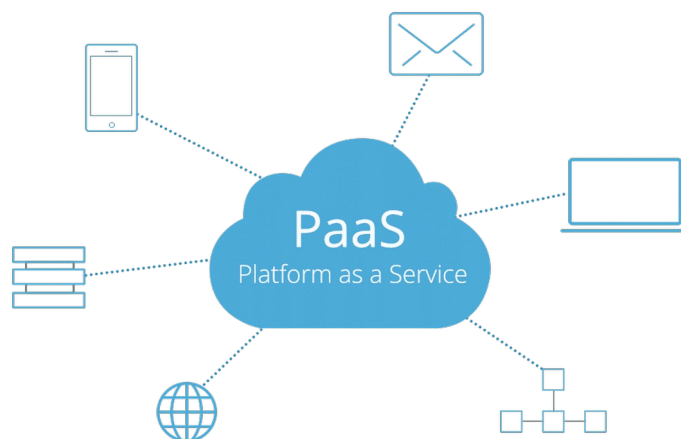
2.5.1 Infrastructure as a Service (IaaS)

Infrastruktura kao servis predstavlja najosnovniju kategoriju usluga računarstva u oblaku. Sa IaaS-om se iznajmljuje IT infrastruktura - serveri i virtualne mašine (VM), skladišta, mreže, operativni sistemi - od dobavljača oblaka po principu „pay-as-you-go“ odnosno plati koliko potrošiš [3].



2.5.2 Platform as a Service (PaaS)

Platforma kao usluga odnosi se na usluge računarstva u oblaku koje pružaju okruženje na zahtev za razvoj, testiranje, isporuku i upravljanje softverskim aplikacijama. PaaS je dizajniran da programerima olakša brzo kreiranje veb ili mobilnih aplikacija, bez briga o postavljanju ili upravljanju osnovnom infrastrukturom servera, skladišta, mreže i baza podataka potrebnih za razvoj [3].



2.5.3 Serverless computing

Preklapajući se sa PaaS-om, računarstvo u oblaku bez servera fokusira se na izgradnju funkcionalnosti aplikacije bez trošenja vremena na kontinuirano upravljanje serverima i infrastrukturom potrebnom za to. Dobavljač usluga u oblaku (*cloud provider*) obrađuje podešavanja, planiranje kapaciteta i upravljanje serverom umesto korisnika. Arhitekture bez servera su visoko skalabilne i vode se događajima, samo koristeći resurse kada se pojavi određena funkcija ili okidač [3].



2.5.4 Software as a Service (SaaS)

Softver kao usluga je metoda za isporuku softverskih aplikacija putem Interneta, na zahtev i obično na osnovu pretplate. Sa SaaS-om, dobavljači usluga u oblaku hostuju i upravljaju softverskom aplikacijom i osnovnom infrastrukturom i bave se bilo kakvim održavanjem, poput nadogradnje softvera i zakrpe za bezbednost. Korisnici se povezuju sa aplikacijom preko Interneta, obično pomoću veb pretraživača na svom telefonu, tabletu ili računaru [3].



2.6 Database as Service (DBaaS)

2.6.1 Karakteristike

Jedna od najbrže rastućih kategorija softverskih usluga (SaaS), kada je reč o bazama podataka na *cloud*-u je *Database-as-a-Service* (DBaaS) rešenje. DBaaS omogućava korisnicima da, putem *cloud*-a, pristupaju i koriste bazu podataka, bez potrebe za posedovanjem svog hardvera, kao što su serveri i skladišta, i bez instalacije ikakvog softvera i održavanja same baze, za šta je zadužen *cloud* provajder [6]. Prilikom rada sa bazom na *cloud*-u, korisnik (aplikacija) ima iste mogućnosti kada se radi o manipulaciji nad podacima potpuno identično sa standardnom bazom koja je podignuta na lokalnoj mreži. Takođe, *cloud* baze moraju da obezbede konstantnu dostupnost (**reliability**), zaštitu (**security**), kao i mehanizme za oporavak u slučaju otkaza hardvera (**recovery**). DBaaS baze su veoma dobro rešenje za kompanije manjih i srednjih veličina, tako što održavanje i administracija baze prelazi na odgovornost provajdera, što znatno utiče na smanjenje troškova održavanja.



Database as a Service

Jednostavnost (**simplicity**) je reč kojom se opisuje velika popularnost DBaaS-a. Proces *database provisioning*-a kod DBMS-a zahteva znatno duže vreme trajanja izvršenja, nego *auto-provisioning* koji pojednostavljuje taskove. Niska cena (**low cost**) je još jedna atraktivna tačka za prelazak na DBaaS, jer se plaća samo upotreba. Skalabilnost (**scalability**) je takođe važna karakteristika zbog koje se korisnici odlučuju za prelazak na DBaaS, iz razloga što ovaj sistem omogućava povećanje trenutnih resursa u skladu sa zahtevima. Pored svega navedenog, DBaaS ima funkciju

da služi podacima sa različitih strana sveta pružajući ulazne tačke, što će biti korisno kupcima za pristup podacima bez odlaganja [7].

2.6.2 Provajderi

Trenutno su glavni provajderi ovog tipa servisa Amazon i Microsoft Azure. Amazon nudi različite RDS *database engine*-e, što je potrebno odabrati zavisno od aplikacije, a to su Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, MicrosoftSQL Server. Sa druge strane Microsoft Azure obezbeđuje Azure Database za PostgreSQL, MySQL i MariaDB [7]. Dosta različitih *cloud database* servis provajdera obezbeđuju DBaaS koji se dalje deli na neke od tri velikih kategorija, a to su relacione baze podataka, nerelacione baze podataka i operativne virtuelne mašine na kojima trči instanca lokalne baze podataka kao što je SQL [9].

2.6.3 SQL i NoSQL

Od baza koje su dostupne na *cloud*-u, neke su SQL bazirane, dok neke koriste NoSQL model.

Relacione baze, koje su bazirane na strukturnom jeziku upita (SQL), su vezane za koncept tabele i ovakav tip baza je idealan kada se radi o struktuiranim podacima i biznis transakcijama kao i podacima gde se traži visok nivo konzistentnosti. Razlog je taj da se relacione baze strogo pridržavaju ACID svojstava kada je reč o transakcijama.

NoSQL (*Not only SQL*) baze podataka nisu zasnovane na tabelarnom pamćenju podataka, što je glavna razlika u odnosu na relacione. Ove baze obezbeđuju fleksibilnu šemu za pamćenje sadržaja. U zavisnosti od baze, podaci mogu biti pamćeni u obliku dokumenta, ključ-vrednost parova, grafova itd. Ovakve baze su pogodne za pamćenje velike količine nestruktuiranih ili polustruktuiranih podataka. Primeri SQL i NoSQL *cloud* baza dati su na sledećoj slici [10].

	Virtual Machine Deployment	Database as a Service
SQL Data Model	<ul style="list-style-type: none"> • EDB Postgres Advanced Server • IBM DB2^[10] • Ingres (database)^[11] • MariaDB^{[12][13]} • MySQL^[14] • Nuodb^[15] • Oracle Database^{[16][17]} • PostgreSQL^[18] • SAP HANA^{[19][20][21]} 	<ul style="list-style-type: none"> • Amazon Aurora, MySQL based service • Amazon Relational Database Service • Clustrix Database as a Service^[22] • Crunchy Bridge[®], PostgreSQL as a Service^[23] • EnterpriseDB Postgres Plus Cloud Database^[24] • Google Cloud SQL^[25] • Heroku PostgreSQL as a Service (shared and dedicated database options)^[26] • MariaDB SkySQL[®]^[27] • Microsoft Azure SQL Database (MS SQL)^[28] • Oracle Database Cloud Service^[29] • Snowflake Cloud Data Warehouse • Xeround Cloud Database* - MySQL front-end (*service no longer available)^[30]
NoSQL Data Model	<ul style="list-style-type: none"> • Apache Cassandra on Amazon EC2^[31] or Google Compute Engine^[32] • ArangoDB on Amazon EC2, Google Compute or Microsoft Azure^[33] • Clusterpoint Database Virtual Box VM^[34] • CouchDB on Amazon EC2^[35] or Google Cloud Platform^[36] • EDB Postgres Advanced Server • Hadoop on Amazon EC2,^[37] Google Cloud Platform,^[38] or Rackspace^[39] • MarkLogic on Amazon EC2^[40] or Google Cloud Platform^[41] • MongoDB on Amazon EC2,^[42] Google Compute Engine,^[43] Microsoft Azure,^[44] or Rackspace^[45] • Neo4j on Amazon EC2^[46] or Microsoft Azure^[47] 	<ul style="list-style-type: none"> • Amazon DynamoDB • Amazon SimpleDB^[citation needed] • Azure Cosmos DB • Cloudant Data Layer^[48] (CouchDB) • EnterpriseDB Postgres Plus Cloud Database^[24] • Google Cloud Bigtable^[49] • Google Cloud Datastore^[50] • MongoDB Database as a Service (several options)^[51] • Oracle NoSQL Database Cloud Service^[52] • Amazon DocumentDB

2.7 Prednosti i nedostaci *cloud* baza podataka

Evo nekoliko **prednosti** za baze podataka na oblaku [8, 11]:

- Cena održavanja
- Skalabilnost
- Elastičnost
- Smanjen administrativni teret
- Dostupnost
- Poboljšana bezbednost
- Jednostavna implementacija
- Efikasan oporavak

Što se tiče **nedostataka** navode se sledeći [8, 11]:

- Cena održavanja
- Nefleksibilnost
- Vreme zastoja
- Redudantnost
- Propusni opseg

2.7.1 *Cena održavanja*

Cena održavanja (*cost*) se navodi i kao prednost i nedostatak iz razloga što može znatno uštedeti novac kompaniji ukoliko koristi na primer jedan ili mali broj servisa, čime oslobađa IT osoblje od održavanja i ne mora imati svoj *data centar*. Sa druge strane, u slučaju skaliranja korišćenje servisa na oblaku može biti znatno skupo [8].

Prednosti

2.7.2 *Skalabilnost*

Važna prednost *cloud* baza jeste efikasnija **skalabilnost** (*scalability*), tj. mogućnost da, kako raste količina podataka koji se pamte, sistem automatski ili na zahtev angažuje nove čvorove za pamćenje istih. Kod DBaaS baza, skalabilnost se ostvaruje brže, jeftinije i efikasnije (uz par klikova) u odnosu na tradicionalne baze, kod kojih bi bila neophodna kupovina i podizanje novih servera [9]. U poređenju sa pre nekoliko godina, baze podataka hostovane na oblaku mogu se skalirati brzo, jeftino i efikasno. Skaliranje po bilo kojoj dimenziji obično zahteva dodavanje ili oduzimanje čvorova od klastera da bi se promenio kapacitet skladištenja, I / O operacije u sekundi (IOPS) ili ukupni proračun dostupan kako bi se izvršili upiti. Ta operacija, naravno, zahteva ponovnu distribuciju kopija podataka i slanje između čvorova [8].

2.7.3 *Elastičnost*

DBaaS takođe obezbeđuje svojstvo **elastičnosti** (*elasticity*) što podrazumeva automatsko obezbeđivanje dodatnih resursa kada imamo veće opterećenje sistema (veliki broj upita ili veliki broj redova) [9].

2.7.4 Smanjen administrativni teret

Baza podataka sa manje funkcija je manje fleksibilna - ali je mnogo lakša za upravljanje odnosno ima **smanjen administrativni teret** (*reduced administrative burden*). Baza podataka koja se hostuje na oblaku, uglavnom se samostalno upravlja, ne eliminiše administratora baze podataka, ali može ukloniti nepotrebne funkcije koje obično troše mnogo vremena i napora DBA-e (*Database Administrator*). To omogućava da administrator svoje vreme fokusira na važnija pitanja. Može čak i da omogućiti kompaniji da radi bez namenskog (*part-time*) ili stalnog (*full-time*) DBA, izbegavajući specijalizaciju koja može dovesti do uskih grla u timu [8].

2.7.5 Dostupnost

Dodatna prednost *cloud* baza jeste činjenica što je omogućena veća **dostupnost** (*accessibility*). Za pristup *cloud* bazi jedino što je potrebno jeste pristup Internetu, što praktično znači da je baza postala bilo kad i bilo gde dostupna. Ovo svojstvo stalne dostupnosti baze olakšava komunikaciju između korisnika na različitim lokacijama, čini bazu lakom za upotrebu, a takođe utiče i na povećanje produktivnosti [11].

2.7.6 Poboljšana bezbednost

Ukoliko je baza podataka podignuta u lokalu veoma je važno razmisliti o **bezbednosti** i unaprediti je na najbolji mogući način (*improved security*). Uvek treba držati *kernel* baze podataka i ostale kritične softvere na najnovijim verzijama tj. ažurnim kako bi se izbegle najnovije digitalne pretnje. Problem je što većina kompanija ili to radi loše ili ne radi uopšte, tako da prelaskom na *cloud* baze podataka osigurana je bezbednost kompletne infrastrukture [8].

2.7.7 Jednostavna implementacija

Cloud hosting omogućava preduzećima da zadrže iste aplikacije i poslovne procese bez potrebe da se bave pozadinskim tehnikama. Pristup i upravljanje odnosno **implementacija** se vrši putem Interneta, tako da se infrastrukturi *cloud*-a može pristupiti lako i brzo (*easy implementation*) [11].

2.7.8 Efikasan oporavak

Računarstvo u oblaku omogućava brže i tačnije pronalaženje aplikacija i podataka sa što manje zastoja jer to predstavlja najefikasniji plan **oporavka** (*efficient recovery*) [11].

Nedostaci

2.7.9 Nefleksibilnost

Mnogi problemi sa skalabilnošću i performansama baze podataka mogu se rešiti samo softverskom zakrpom (*patch*), nadogradnjom (*upgrade*) ili podešavanjem konfiguracije (*configuration setting*). Međutim, zaključana priroda baze podataka u

oblaku može zabraniti upotrebu ovih rešenja što se opisuje **nefleksibilnošću** (*inflexibility*). Rešenje: Ako ne postoje jedinstveni zahtevi za bazom podataka - a većina ljudi koji misle da zaista nemaju - verovatno imaju mnogo veću fleksibilnost nego što im je potrebna. Preporučuje se savet od drugih, bilo da se radi o vršnjacima ili stručnoj konsultantskoj firmi [8].

2.7.9 Vreme zastoja

Iako nije toliko zabrinjavajuće kao nekada, računarstvo u oblaku ima reputaciju povremenih **velikih prekida** (*downtime*). Rešenje: Izgraditi infrastrukturu u više regiona, zona dostupnosti, dobavljača i centara podataka kako bi se umanjili rizici. Izgradnja na jednoj tački neuspeha trebali bi biti gotovi - čak i za vrlo mlade *startup*-ove [8].

2.7.10 Redudantnost

Cloud server nije redudantan tako da tehnologija vremenom može otkazati, što znači da treba uvek imati u vidu neki plan **redudantnosti**. Iako je to dodatni trošak, u većini slučajeva će se isplatiti. Takođe *cloud* baze obezbeđuju i automatske *backup*-ove i replike podataka smeštene na druge geo lokacije, za slučaj da dođe do pada sistema. Zbog svega ovoga postavlja se pitanje ko sve može imati pristup podacima korisnika, tj. može doći do narušavanja privatnosti istih. Za neke kompanije, ovo može biti u nesaglasnosti sa propisanim bezbednosnim polisama [11].

2.7.11 Propusni opseg

Problemi s **propusnim opsegom**. Za idealne performanse, klijenti moraju da planiraju u skladu sa tim da ne pakuju velike količine servera i uređaja za skladištenje u mali skup centara podataka [11].

3. Praktični deo – ElephantSQL

3.1 Uopšteno o tehnologiji

ElephantSQL predstavlja PostgreSQL *open-source* bazu podataka hostovanu na *cloud*-u. Provajder ovog servisa upravlja administrativnim procesima vezanim za PostgreSQL kao što su: instalacija, nadogradnja do najnovije stabilne verzije, kao i upravljanje backup-ovima odnosno rezervnim kopijama. ElephantSQL je takođe moguće integrisati sa nekoliko *cloud* aplikacionih platformi, što omogućava da korisnička aplikacija i baza mogu da se nađu na istom data centru. ElephantSQL podržava sledeće *cloud* platforme: *Amazon Web Services (AWS)*, *Google Cloud Platform*, *Microsoft Azure* i *IBM Cloud* [12].



Neke od karakteristika ElephantSQL-a su [12]:

- konstantni monitoring nad serverima
- backup na dnevnom nivou
- mogućnost da se prilikom hardverskog ili softverskog otkaza, obrada automatski prebaci na drugi server (*Follower*) čija lokacija može biti i na drugom *cloud* provajderu
- web browser alat za kreiranje SQL upita, što otklanja potrebu za instaliranjem alata poput pgAdmin-a ili psql-a

3.2 Planovi baze podataka, cene i servisi

Planovi

Prilikom zahteva za novom instancom servisa, korisnik specificira na koji plan korišćenja želi da se pretplati. Kao i većina DBaaS-a, korisnik plaća servis po iskorišćenosti, gde se u slučaju ElephantSQL-a radi o vremenskoj dostupnosti servisa, i plaća se na kraju svakog meseca.

Planovi koje ElephantSQL nudi korisnicima su [13]:





- Deljivi planovi (*Shared Plans*)
- Namenski planovi (*Dedicated Plans*)
- Namenski planovi sa opcijom sledbenika (*Dedicated Plans with Followers*)

3.2.1 Deljivi planovi

Shared Plans daje jednu postavljenu PostgreSQL bazu podataka sa automatski konfigurisanim parametrima baze podataka, hostovanu na *cloud*-u. Za hobi projekte i razvoj preporučuje se zajednički plan, budući da server baze podataka opslužuje više korisnika. Planovi u okviru ove grupe se razlikuju po tome, koliko je prostora na

raspolaganju korisniku, kao i po tome koliko konkurentnih konekcija na instanci, plan podržava [13].





Shared Instances



TINY TURTLE	SIMPLE SPIDER	CRAZY CAT	PRETTY PANDA
Shared high performance server	Shared high performance server	Shared high performance server	Shared high performance server
20 MB data	500 MB data	1 GB data	2 GB data
5 concurrent connections	10 concurrent connections	15 concurrent connections	20 concurrent connections
			
FREE	\$ 5	\$ 10	\$ 19
GET STARTED	PER MONTH	PER MONTH	PER MONTH
Try now for FREE	Try now for \$5/month	Try now for \$10/month	Try now for \$19/month

3.2.2 Namenski planovi

Baza se smešta na poseban server koji je dodeljen isključivo konkretnom korisniku. Korisnik može da specificira *cloud* provajdera tj. *data centar* na kome želi da se nalazi njegova baza. Takođe, ovi planovi obezbeđuju znatno više raspoloživog prostora, veći broj konkurentnih konekcija, veći broj mogućnosti kada je reč o monitoringu kao i mogućnost da jedna instanca sadrži veći broj baza [13].

Dedicated Instances

BLISSFUL BUTTERFLY	HAPPY HIPPO	ENORMOUS ELEPHANT	PUFFY PIGEON
2 GB RAM	4 GB RAM	8 GB RAM	16 GB RAM
50 GB data	100 GB data	250 GB data	500 GB data
50 concurrent connections	Hundreds of connections	Hundreds of connections	Hundreds of connections
			
\$ 49	\$ 99	\$ 199	\$ 399
PER MONTH	PER MONTH	PER MONTH	PER MONTH
Try now for \$49/month	Try now for \$99/month	Try now for \$199/month	Try now for \$399/month

DRAMATIC DOLPHIN	RUTHLESS RAT
32 GB RAM	64 GB RAM
1000 GB data	2000 GB data
Hundreds of connections	Hundreds of connections
	
\$ 749	\$ 1 399
PER MONTH	PER MONTH
Try now for \$749/month	Try now for \$1 399/month

3.2.3 Namenski planovi sa opcijom sledbenika

Ova grupa planova pruža iste opcije kao i namenski planovi, s tim što obezbeđuje dodatne mogućnosti koje se tiču bržeg oporavka baze kroz održavanje *read only* kopija baze (*follower-i*), koje ne samo da ubrzavaju upite čitanja, već služe i kao *stand-by*, spremne da preuzmu ulogu glavne baze, ukoliko dođe do pada instance. Nadgledanje klastera se vrši putem redundantnih monitora koji upravljaju otkazivanjem ukoliko dođe do padanja glavne instance. *Follower* može biti lociran i na nekom drugom geografskom regionu u odnosu na lokaciju glavne instance ili čak kod *3rd party* dobavljača usluga tj. na drugom *cloud* provajderu [13].

Dedicated Instances with Follower


BUTTERFLY WITH FOLLOWER

2 dedicated servers

2 GB RAM

50 GB data

50 concurrent connections



\$ 98

PER MONTH

Try now for \$98/month


HIPPO WITH FOLLOWER

2 dedicated servers

4 GB RAM

100 GB data

Hundreds of connections



\$ 198

PER MONTH

Try now for \$198/month


ELEPHANT WITH FOLLOWER

2 dedicated servers

8 GB RAM

250 GB data

Hundreds of connections



\$ 398

PER MONTH

Try now for \$398/month


PIGEON WITH FOLLOWER

2 dedicated servers

16 GB RAM

500 GB data

Hundreds of connections



\$ 798

PER MONTH

Try now for \$798/month


DOLPHIN WITH FOLLOWER

2 dedicated servers

32 GB RAM

1000 GB data

Hundreds of connections



\$ 1 598

PER MONTH

Try now for \$1 598/month


RAT WITH FOLLOWER

2 dedicated servers

64 GB RAM

2000 GB data

Hundreds of connections



\$ 2 798

PER MONTH

Try now for \$2 798/month

Servisi

Uz svaki od planova uključeni su sledeći servisi [13]:

- Nadgledanje (*Monitoring*)
- Rezervni kopije (*Backups*)
- Administracija (*Administration*)
- Mobilnost (*Mobility*)
- Naplate (*Billing*)
- Održavanje (*Support*)

3.2.4 Nadgledanje

Svi serveri se kontinuirano nadgledaju radi dostupnosti i performansi. Najčešće se otkrivaju problemi i rešavaju ih pre nego što postanu problem.

3.2.5 Rezervne kopije

Automatske rezervne kopije (*pg_dump*) izvode se svakodnevno. Potpune rezervne kopije servera (osnovna rezervna kopija) izvode se jednom nedeljno, WAL sigurnosne kopije se vrše na svaka dva minuta.

3.2.6 Administracija

Sve instance imaju pristup veb pregledaču ElephantSQL, što je sjajno za jednostavnu inspekciju i za testne upite.

3.2.7 Mobilnost

ElephantSQL je dostupan u rastućem skupu IaaS. Ovo omogućava lako premeštanje i distribuciju podataka širom sveta.

3.2.8 Naplate

Sve planove naplaćuju drugi, tako da možete isprobati čak i najveće tipove instanci za samo nekoliko penija. Naplata se vrši na kraju svakog meseca i naplaćuje se samo za vreme kada je instanca bila dostupna.

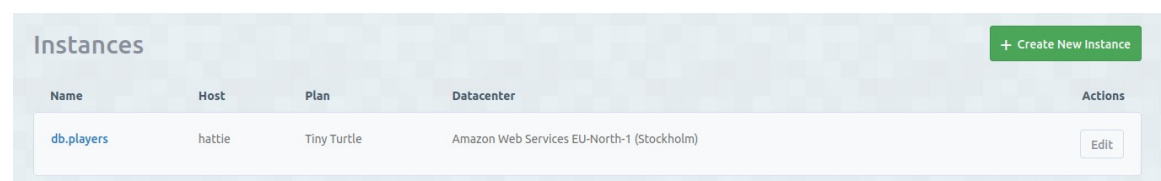
3.2.9 Korisnička podrška

Pružaju korisničku podršku odgovaranjem na mail 24/7, a za veće planove i telefonsku podršku. Takođe obezbeđuju i dogovor ukoliko potencijalni klijent želi da razgovara o tome kako ElephantSQL može da podrži njegovo poslovanje.

3.3 Kreiranje instance i navigacioni meni

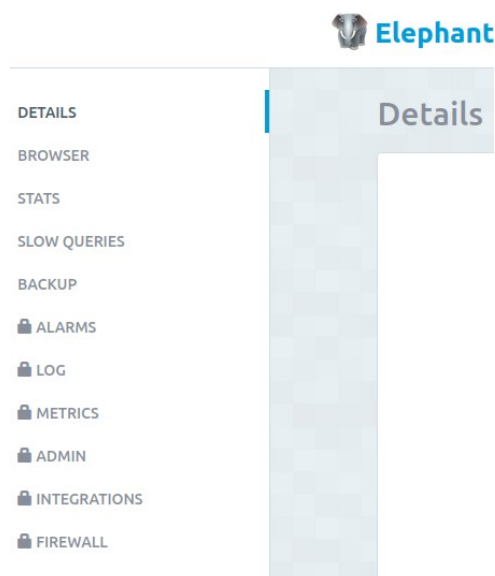
Za potrebe demonstriranja DBaaS-a, u primerima je korišćena ElephantSQL baza podataka, pri čemu je kreirana instanca sa deljivim planom *Tiny Turtle*, koji je besplatan, ali ograničen obzirom da je slobodan prostor koji je korisniku na raspolaganju samo 20 MB. Pritom dozvoljene su pet konkurentne konekcije sa instancom u jednom trenutku.

Prilikom kreiranja instance, korisnik podešava data centar i region gde želi da podigne instancu (deljivi planovi imaju manji izbor data centar-a), s tim da ukoliko se korisnička aplikacija koja komunicira sa bazom, nalazi na nekoj *cloud* platformi, poželjno je zbog latentnosti, da se i baza i aplikacija nalaze na istom data centru [12].

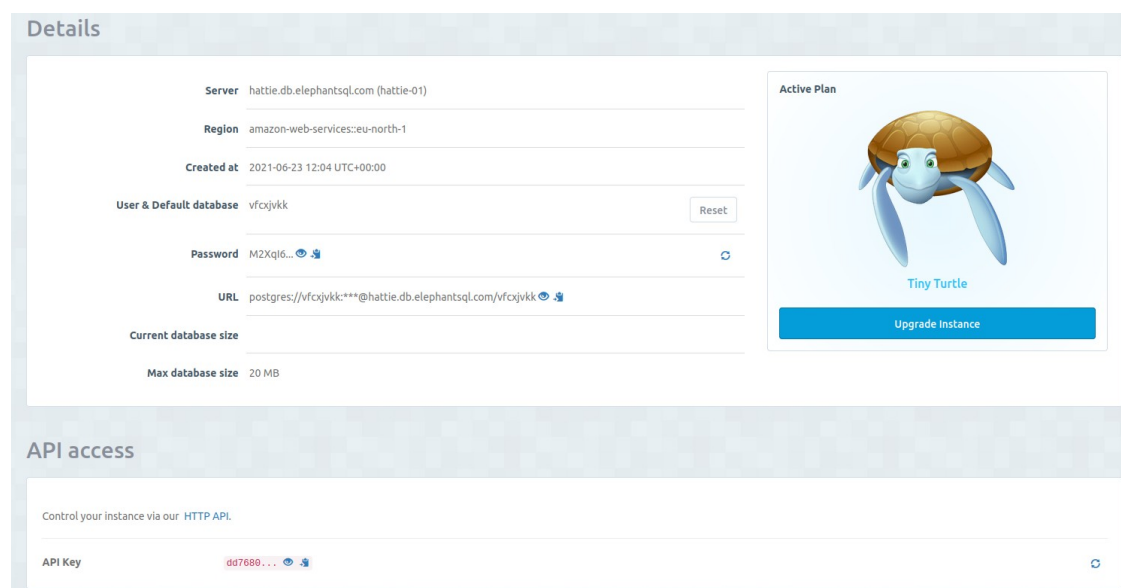


Name	Host	Plan	Datacenter	Actions
db.players	hattie	Tiny Turtle	Amazon Web Services EU-North-1 (Stockholm)	Edit

Sa leve strane prozora u pretraživaču se može uočiti navigacioni meni odnosno *sidebar*, gde su na raspolaganju sledeće opcije: detalji, pretraživač, statistika, upiti, rezervna kopija, kao i alarmi, logovi, metrike, administracioni panel, integracije i zaštitni zid, što je ilustrovano na slici ispod.



Nakon kreiranja instance, u *Details* prozoru, korisnik može da vidi sledeće informacije: server i region na kome je instanca, vreme kreiranja instance, plan pretplate, šifre, pristupni URL, iskorišćenost datog prostora i maksimalna veličina prostora za dati plan. Takođe je omogućen pristup preko API-ja. Na slici ispod dat je prikaz *Details* tab-a ElephantSQL web konzole, na kojoj se vide detalji kreirane instance [12].

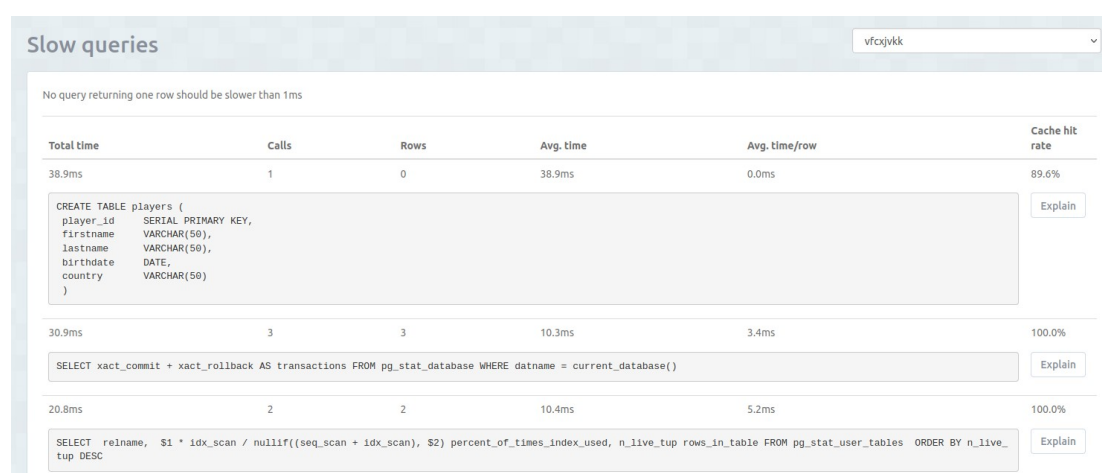


3.4 Interakcija sa bazom

Za interakciju sa bazom, korisniku je na raspolaganju ElephantSQL web konzola, ali je pristup moguć i uz pomoć standardnih alata kao što su *pgAdmin* ili *psql* interaktivna konzola.

3.4.1 Web konzola

Link do ElephantSQL web stranice se može naći na upravljačkoj konzoli instance. Web konzola ElephantSQL-a pruža razne alate za: kreiranje SQL upita, nadgledanje instanci, za pravljenje backup-a i follower-a, uvid u razne statistike koje se tiču izvršavanja upita, itd [12].



Total time	Calls	Rows	Avg. time	Avg. time/row	Cache hit rate
38.9ms	1	0	38.9ms	0.0ms	89.6%
<pre>CREATE TABLE players (player_id SERIAL PRIMARY KEY, firstname VARCHAR(50), lastname VARCHAR(50), birthdate DATE, country VARCHAR(50))</pre>					
30.9ms	3	3	10.3ms	3.4ms	100.0%
<pre>SELECT xact_commit + xact_rollback AS transactions FROM pg_stat_database WHERE datname = current_database()</pre>					
20.8ms	2	2	10.4ms	5.2ms	100.0%
<pre>SELECT relname, \$1 * idx_scan / nullif((seq_scan + idx_scan), \$2) percent_of_times_index_used, n_live_tup rows_in_table FROM pg_stat_user_tables ORDER BY n_live_tup DESC</pre>					

Na slici iznad dat je primer kreiranja test tabele *players* [14], pri čemu je upit već izvršen, a zatim na slici ispod popunjavanje tabele uz pomoć ElephantSQL Browser-a.



```
INSERT INTO players (firstname,lastname,birthdate,country)  
VALUES ( 'Paul', 'Pogba', '1993-03-15', 'France');  
INSERT INTO players (firstname,lastname,birthdate,country)  
VALUES ( 'Casey', 'Short', '1990-08-23', 'United States');
```

Table queries ▾ Previous queries ▾

No rows returned

Na prvoj slici ispod je prikazan izvršeni upit koji vraća sve igrače iz tabele, dok je na drugoj izvršen upit koji vraća ime i prezime igrača čiji je *id* jednak vrednosti dva.

SQL Browser vfcxjvkk ✓ Query completed

```
SELECT *
FROM players;
```

Table queries ▾ Previous queries ▾ Execute ▶

player_id	firstname	lastname	birthdate	country
1	Paul	Pogba	1993-03-15	France
2	Casey	Short	1990-08-23	United States

SQL Browser vfcxjvkk ✓ Query completed

```
SELECT firstname, lastname
FROM players
WHERE player_id = 2;
```

Table queries ▾ Previous queries ▾ Execute ▶

firstname	lastname
Casey	Short

3.5 Mogućnosti i alati

Preko ElephantSQL konzole korisnik ima jednostavan pristup mnogim opcijama koje nisu dostupne kod klasičnih alata za pristup bazi. U ove opcije spadaju:

- Praćenje raznih statistika vezanih za performanse izvršavanja upita
- Jednostavan način na uticanje skalabilnosti sistema
- Laka manipulacija *backup*-om
- Praćenje performansi servera uz pomoć dijagrama

Treba imati u vidu da su neki od ovih alata zaključani za besplatne planove, tj. dostupni su samo uz planove koji se naplaćuju.

Preko *Stats tab*-a u okviru ElephantSQL *web* konzole, korisnik ima uvid u statistiku koja se tiče izvršavanja upita, kao i informacije o trenutno aktivnim konekcijama sa instancom. Na slici ispod dat je prikaz statistike. Pored informacije o verziji PostgreSQL-a koja se koristi, ovaj tab prikazuje statistike vezane za iskorišćenost indeksa kod pretraživanja podataka. Statistika *Index hit rate* pokazuje koliki procenat indeksa koji se koriste, se nalazi u baferu glavne memorije. bez pristupa disku, dok *Cache hit rate* daje informacije o generalnom procentu pogodaka

kod pribavljanja podataka. *Index Usage* pokazuje, za svaku tabelu, kolika je u procentima iskorišćenost indeksa prilikom pretraga tih tabela. Ukoliko je ovaj broj mali, a tabela sadrži veliki broj redova, poželjno je kreirati indekse nad poljima nad kojima se često vrši pretraga. Takođe statistika pokazuje i kreirane, ali neiskorišćene indekse, čije bi se uklanjanje moglo razmotriti, da bi se ubrzale operacije *update*-a nad tom tabelom. U ovom prozoru se takođe mogu videti i trenutno otvorene konekcije sa instancom, a po potrebi se mogu i zatvoriti.

Database statistics

PostgreSQL version: 11.9 (Ubuntu 11.9-1.pgdg20.04+1)

TPS: 0.0012

Index hit rate: 71.4% (Should be 99%, else add indexes)

Cache hit rate: 75.0% (Should be 99%, else upgrade to a larger instance)

Index usage (Should be 99% for every table with more than 10 000 rows)

Relname	Percent of times index used	Rows in table
players	50	2

Unused indexes (Could be removed and increase insert performance)

No data available

Connections

1 currently open

Since	Application name	Client address	State	Query
2021-06-26T10:30:11.076Z	elephantsql-stream	18.234.231.5	active	SELECT xact_commit + xact_rollback AS transactions FROM pg_stat_database WHERE datname = current_database()

Korisnik može da preko *tab*-a *Slow Queries* ima uvid u vreme koje je bilo potrebno da bi se realizovali određeni upiti. Ovaj prozor omogućava identifikaciju upita koji se sporo izvršavaju. Ovakvi neefikasni upiti mogu znatno da utiču na performanse aplikacije koja koristi bazu. Na slici ispod dat je prikaz *Slow Queries tab*-a. Upiti na slici, čije je vreme izvršenja najduže, su implicitno izvršeni prilikom pregleda *Stats* taba, na osnovu njihovog rezultata je računata statistika vezana za indekse.

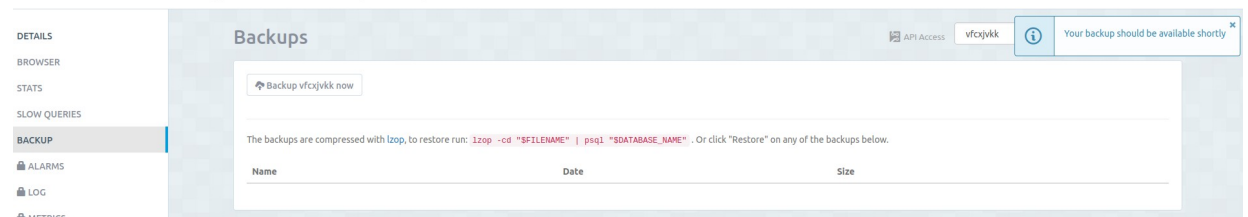
Slow queries

No query returning one row should be slower than 1ms

Total time	Calls	Rows	Avg. time	Avg. time/row	Cache hit rate
144.3ms	14	14	10.3ms	0.7ms	100.0%
SELECT xact_commit + xact_rollback AS transactions FROM pg_stat_database WHERE datname = current_database()					
136.8ms	13	13	10.5ms	0.8ms	100.0%
SELECT relname, \$1 * idx_scan / nullif((seq_scan + idx_scan), \$2) percent_of_times_index_used, n_live_tup rows_in_table FROM pg_stat_user_tables ORDER BY n_live_tup DESC					
8.6ms	14	14	0.6ms	0.0ms	100.0%
<insufficient privilege>					

Kada je reč o *backup*-ovima, ElephantSQL automatski kreira rezervne kopije instanci na dnevnom nivou za planove koji se naplaćuju. Ove rezervne kopije se smeštaju na skladištu u oblaku (*cloud storage*) i uvek su dostupne korisnicima. Za namenske servere *backup* celokupnog servera se izvršava svake nedelje, s tim da se

backup Write ahead Log-a (WAL) prebacuje na *Amazon S3 Object storage* svakog minuta. Na slici ispod je dat prikaz *Backup tab-a*, na kome se mogu ručno kreirati rezervne kopije instanci, kao i učitavati iz fajlova sa sledećim ekstenzijama: *.sql*, *.zip*, *.gz*, *.tar*. Takođe na ovom prozoru se nalaze informacije koje se tiču svih kreiranih *backup-ova* koji su prebačeni na *cloud*. Svaki backup baze, korisnik može preuzeti u obliku kompresovanog fajla, a može i uz pomoć *Restore* opcije odmah učitati isti.



4. Zaključak

Cloud baza podataka je kolekcija struktuiranih ili nestruktuiranih podataka, koji se nalaze na privatnoj ili javnoj *cloud* infrastrukturi. Bazu na *cloud*-u je moguće implementirati preko zakupa virtuelne mašine, na kojoj korisnik sam podiže odgovarajuću instancu ili preko baze u vidu servisa (DBaaS) i plaća korišćenje usluga koje obezbeđuje provajder. Ovim putem se gubi potreba za posedovanjem adekvatnog hardvera kao i instaliranja i održavanja baze. Prednosti DBaaS-a su: ušteda u troškovima – korisnik ne mora da ima infrastrukturu, da plaća softver i osoblje koje bi održavalo ceo sistem, efikasna skalabilnost, kako vertikalna tako i horizontalna, veća dostupnost u poređenju sa tradicionalnim bazama (neophodna je samo internet konekcija), ubrzan razvoj aplikacija, jednostavno upravljanje bazama (putem *web browser* konzole) i dr.

U ovom radu je kao primer DBaaS-a obrađen ElephantSQL *hosting* servis za PostgreSQL bazu. Ovaj servis automatizuje ceo proces postavljanja i izvršavanja PostgreSQL klastera i dostupan je na većini većih *cloud* provajdera širom sveta, kao što su *Amazon*, *Google* i *Microsoft*. ElephantSQL ima široku ponudu kada je reč o servisima, počevši od baza na zajedničkom serveru (pogodne za manje projekte), sve do velikih baza, koje se nalaze na više namenskih servera (pogodne za veće kompanije koje insistiraju na boljim performansama). U slučaju hardverskog i softverskog otkaza glavnog čvora, obezbeđeno je automatsko prebacivanje opterećenja na redundantne čvorove – *follower*-e, s tim da ovi čvorovi mogu da se nalaze i na drugim geografskim regijama, kao i na drugim provajderima. ElephantSQL obezbeđuje i efikasan sistem automatskog, dnevnog, *backup*-ovanja baza sa servera na *cloud* skladište koje je korisniku stalno dostupno. Takođe je korisniku na raspolaganju i *point-in-time* oporavak, u koliko je potrebno vratiti bazu u neko prethodno stanje. Još jedna bitna funkcionalnost obezbeđena od ElephantSQL jesu alati za praćenje baze, gde korisnik ima uvid u performanse servera (CPU i memorijska iskorišćenost), koji upiti zahtevaju najviše vremena, uvid u to da li je potrebno kreiranje ili uklanjanje indeksa radi ubrzavanja pretraga, kao i informacije o otvorenim konekcijama sa bazom. Interakcija sa bazom je moguća uz pomoć *user-friendly browser* konzole, ali i preko standardnih alata.

5. Literatura

- [01] Ramez Elmasri and Shamkant Navathe. (2010). Fundamentals of Database Systems (6th. ed.). Addison-Wesley Publishing Company, USA.
- [02] Jain, S., & Alam, M. A. (2017). Comparative Study of Traditional Database and Cloud Computing Database. *International Journal of Advanced Research in Computer Science*, 8(2).
- [03] <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#benefits>
- [04] https://en.wikipedia.org/wiki/Cloud_database
- [05] <https://labs.sogeti.com/cloud-database-dbaas-database-as-a-service/>
- [06] <https://www.ibm.com/cloud/learn/dbaas>
- [07] <https://www.supportpro.com/blog/dbms-and-dbaas/>
- [08] <https://tech.co/news/math-can-business-benefit-cloud-database-2015-08>
- [09] Al Shehri, W. (2013). Cloud database database as a service. *International Journal of Database Management Systems*, 5(2), 1.
- [10] https://en.wikipedia.org/wiki/Cloud_database
- [11] <https://www.red-gate.com/simple-talk/databases/sql-server/performance-sql-server/designing-highly-scalable-database-architectures/>
- [12] <https://www.elephantsql.com/docs/index.html>
- [13] <https://www.elephantsql.com/plans.html>
- [14] <https://www.elephantsql.com/blog/databases-for-beginners-what-is-sql-what-is-postgresql.html>

