Elektronski fakultet u Nišu

Katedra za računarstvo

SEMINARSKI RAD

SISTEMI ZA UPRAVLJANJE BAZAMA PODATAKA

Cloud baze podataka i *Database-as-a-service* rešenja na primeru Google Cloud SQL-a

Student:  
Vukadin Drašković, br. indeksa 1613

Sadržaj

[1. Uvod 3](#_Toc171258644)

[2. Računarstvo u oblaku 4](#_Toc171258645)

[2.1. Tipovi računarstva u oblaku 6](#_Toc171258646)

[2.2. Database as a Service 7](#_Toc171258647)

[3. Google Cloud SQL 11](#_Toc171258648)

[3.1. Kreiranje instance Google Cloud SQL-a 14](#_Toc171258649)

[3.2. Kreiranje instance baze podataka 18](#_Toc171258650)

[3.3. Povezivanje aplikacije na Cloud SQL instancu 20](#_Toc171258651)

[4. Zaključak 23](#_Toc171258652)

[5. Literatura 24](#_Toc171258653)

# Uvod

Tradicionalni sistemi baza podataka, koji su zahtevali značajne resurse za održavanje hardvera i softvera, postepeno ustupaju mesto modernim rešenjima zasnovanim na računarstvu u oblaku. Jedno od najznačajnijih dostignuća u ovom domenu su *Cloud baze podataka* i *Database-as-a-Service* (DBaaS) rešenja, koja omogućavaju korisnicima da efikasno upravljaju i analiziraju velike količine podataka bez potrebe za značajnim ulaganjem u samu infrastrukturu. Google Cloud nudi svoje rešenje, Cloud SQL, za upravljanje i korišćenje realcionih baza podataka kao usluge.

Ovaj seminarski rad će za početak objasniti osnovne pojmove i koncepte računarstva u oblaku, a zatim i istražiti koncept Cloud baza podataka i DBaaS rešenja, gde će kao primer biti uzet upravo Google Cloud SQL. Kroz prikaz upotrebe Cloud SQL-a ovaj seminarski rad ima za cilj da *hands-on* pristupom pojasni glavne karakteristike korišćenja i upravljanja jednog DBaaS rešenja.

# Računarstvo u oblaku

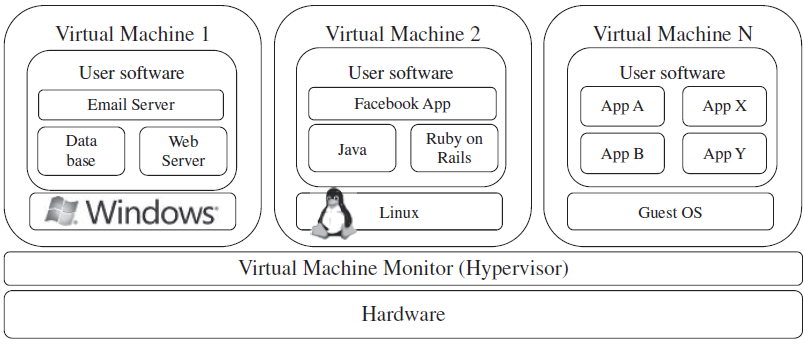
**Računarstvo u oblaku** (engl. *Cloud computing*) je teško definisati jer predstavlja veoma širok pojam i njegove granice nisu jasno definisane [1]. Ipak, veliki broj inženjera i naučnika je pokušao da definiše računarstvo u oblaku i da predstavi njegove glavne karakteristike, a neke od definicija su:

* Računarstvo u oblaku je paralelni i distribuirani računarski sistem, koji se sastoji od većeg broja međusobno povezanih i virtualizovanih računara koji se krajnjem korisniku predstavljaju kao jedan ili više jedinstvenih računara, baziran na SLA (engl. *Service Level Agreements*, srp. *ugovor o nivou usluge*) koji se uspostavlja dogovorom između provajdera i korisnika usluga. - *Rajkumar Buyya*
* Računarstvo u oblaku je veliki skup virtualizovanih resursa koji se lako mogu koristiti, kao npr. hardver, platforme za razvoj sistema ili usluge. Ovi resursi se dinamički mogu skalirati tako da zadovolje novonastale potrebe korisnika, što omogućava i njihovo optimalno korišćenje. Ovaj skup resursa se klijentu nudi u *pay-per-use* (srp. *plati onoliko koliko koristiš*) modelu na osnovu SLA. - *Luis M. Vaquero*
* Naučnici sa Univerziteta Kalifornija u Berkliju su računarstvo u oblaku definisali sa tri glavne karatkeristike: iluzija o postojanju neograničenih resursa, eliminacija unapred ugovorenih obaveza korisnika klauda i mogućnost naplate na osnovu količine korišćenih resursa.

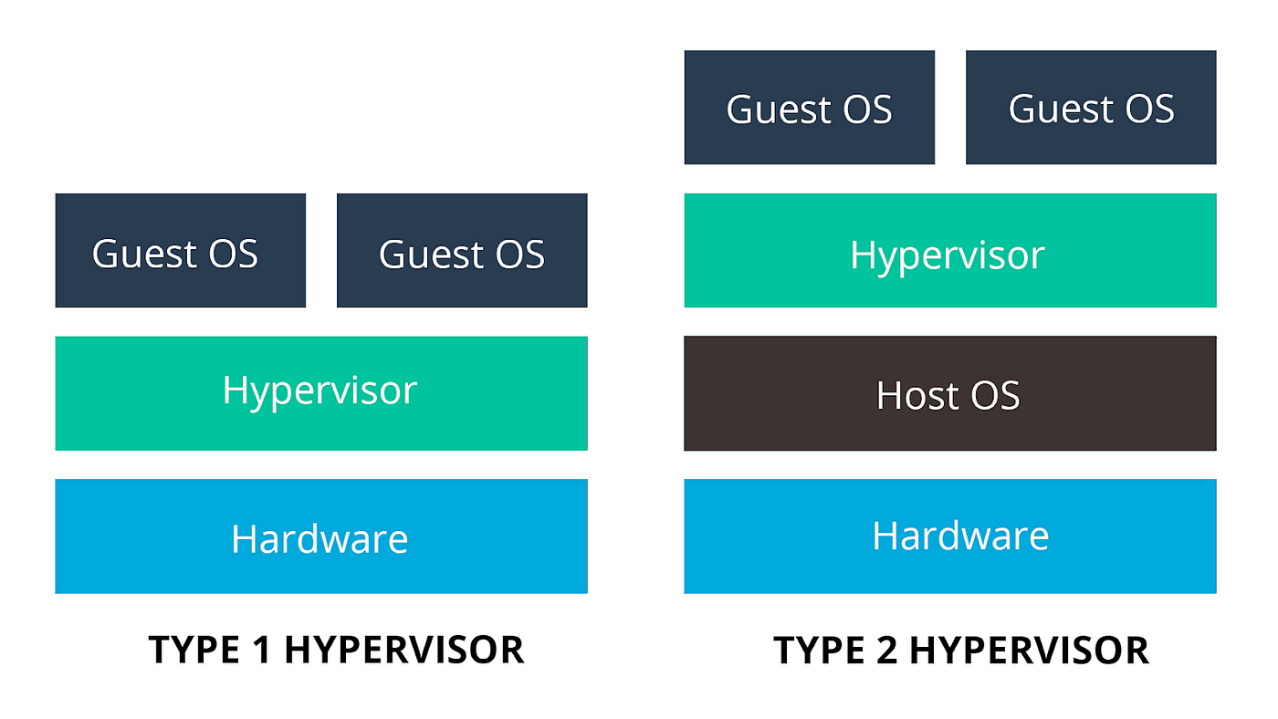
Postoji još veliki broj definicija računarstva u oblaku, ali karakteristike klauda koje sve definicije prepoznaju su:

1. ***pay-per-use*** model
2. **elastičnost** - iluzija o postojanju neograničenih količina resursa
3. ***self-service interface*** - korisnik samostalno koristi resurse bez asistencije provajdera
4. **virtualizovani resursi**

Koncept koji predstavlja osnovu računarstva u oblaku i bez koga klaud ne bi mogao da postoji je **virtualizacija** (engl. *Virtualization*). Ideja o virtualizaciji računarskih resursa, pod kojima se podrazumevaju procesori, UI uređaji, memorije, storidži, itd. postoji već decenijama, ali je sa pojavom klauda doživela ekspanziju. Virtualizacija omogućava pokretanje više operativnih sistema, koji međusobno mogu biti različiti, na jednoj fizičkoj platformi, pri čemu svaki od pokrenutih operativnih sistema predstavlja virtuelnu mašinu (engl. *Virtual machine*) kao što je prikazano na slici 2.1. Fizičku platformu može predstavljati jedan računar, a može predstavljati i skup umreženih računara koji se koriste kao jedan.

  
Slika 2.1. Virtualizacija [1]

Na slici 2.1. je prikazan način funkcionisanja virtualizacije - na najnižem sloju se nalazi hardver koji se treba virtualizovati, tj. nad kojim se treba izvršavati više virtualnih mašina. Softverski nivo koji omogućava virtualizaciju hardvera se zove **Hypervisor**. Hypervisor upravlja pristupom hardveru od strane virtuelnih mašina, kako ne bi došlo do konflikta prilikom pristupanja deljenim resursima. Nad hypervisor-om se instaliraju virtuelne mašine, svaka od njih sa svojim operativnim sistemom, u kojima se mogu pokrenuti raznorazni servisi i usluge koje zahtevaju kijenti.

  
Slika 2.2. Razlika tipa 1 i tipa 2 hypervisor-a [2]

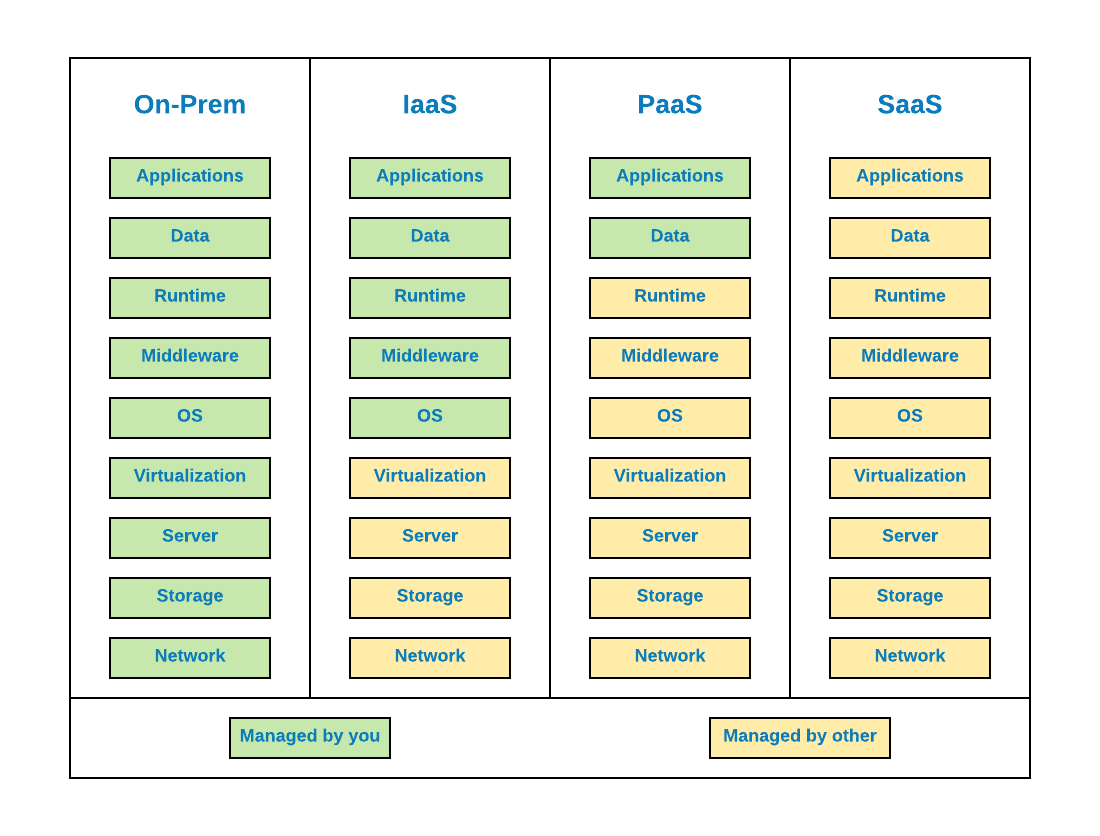
Postoji 3 tipa hypervisor-a:

1. Hypervisor **tipa 0** - Hypervisor se instalira direktno nad hardverom (npr. IBM LPARs).
2. Hypervisor **tipa 1** - specijalizovani softver nalik operativnom sistemu koji ima podršku za virtualizaciju (npr. VMWare ESX).
3. Hypervisor **tipa 1** - operativni sistem opšte namene sa ugrađenim funkcijama za podršku virtualizaciji (npr. Linux with KVM, Microsoft Windows Server with HyperV).
4. Hypervisor **tipa 2** - aplikacije za standardne operativne sisteme koje imaju mogućnost hypervisor-a (npr. VirtualBox)

Na slici 2.2. je prikazana razlika između hypervisor-a tipa 1 i 2.

Virtualizacija potpomaže računarstvo u oblaku jer omogućava da se na zahtev korisnika kreira virtuelna mašina koju će korisnik koristiti i za čije će korišćenjekasnije platiti klaud provajderu. Virtuelne mašine pokrenute na istom računaru se ponašaju kao nezavisni računari i nemaju nikakvu svest o tome da je nad istim skupom resursa koje koristi virtuelna mašina pokrenuta još neka virtuelna mašina.

## Tipovi računarstva u oblaku

  
Slika 2.3. Tipovi računarstva u oblaku [3]

Usluge računarstva u oblaku se grubo mogu podeliti u tri kategorije na osnovu toga koji nivo apstrakcije pruža korisnicima. Ove tri kategorije su prikazane na slici 2.3, a to su:

1. **Infrastruktura kao usluga** (engl. *Infrastructure as a Service*) - IaaS
2. **Platforma kao usluga** (engl. *Platform as a Service*) - PaaS
3. **Softver kao usluga** (engl. *Software as a Service*) - SaaS

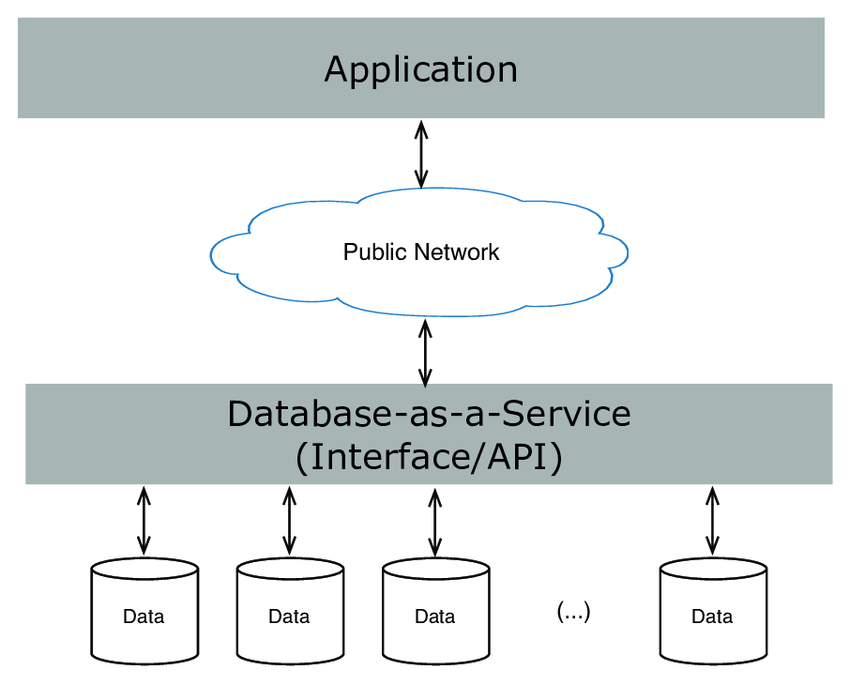
Infrastruktura kao servis je tip računarstva u oblaku koji radi na najnižem nivou apstrakcije. On nudi korisniku usluge korišćenja virtualizovanih resursa, kao što je procesor, memorija, storidža, itd. Korišćenje se uglavnom svodi na odabir operativnog sistema koji će koristiti virtuelna mašina, nad kojim će biti pokrenut određeni softverski stek koji će pružati usluge klijentima.

Za razliku od IaaS koji omogućava korišćenje i virtuelizaciju računarskog hardvera, platforma kao servis nudi viši nivo apstrakcije. Ideja je da se korisniku kao usluga nudi okruženje, odnosno platforma, koju će korinsici koristiti za razvoj aplikacija, njihovo testiranje i isporuku. U ovom tipu računarstva u oblaku, npr. korisniku nije važno koj operativni sistem će aplikacija koristiti, pošto klaud provajder o tim stvarima vodi računa. Jedan od primera PaaS je *Google AppEngine* koji nudi okruženje za razvoj i hostovanje Web aplikacija, koje mogu biti napisane u određenim programskim jezicima, kao npr. *Java* ili *Python*.

Softver kao servis je tip računarstva u oblaku koji nudi najviši nivo apstrakcije. Ideja je da korisnik pristupa ovim uslugama preko Web browser-a, tako da više nema potrebe za instaliranjem softvera na svom računaru. Softver je instaliran i nalazi se na nekom udaljenom računaru, dok korisnik preko internet pretraživača koristi usluge tog softvera. Tipičan primer SaaS modela je *Microsoft Office 365* kome korisnik može koristiti na primer *Word* ili *PowerPoint*, pri čemu nema na svom računaru instalirane te softvere već ih koristi preko internet pretraživača.

## Database as a Service

**Baza podataka kao usluga** (engl. *Database as a Service*, *DBaaS*) je tip računarstva u oblaku, odnosno tip modela softver kao usluga koji krajnjem korisniku nudi mogućnost korišćenja baze podataka bez ikakve potrebe za obezbeđivanjem odgovarajućeg hardvera i instalacijom softvera za bazu podataka [4]. U ovom modelu klaud provajderi upravljaju bazom podataka u potpunosti, što može podrazumevati kreiranje rezervnih kopija i ažuriranje baze podataka. Klaud provajderi takođe vode računa da baza podataka koju koristi klijent bude dostupna i sigurna 24 časa svakog dana u nedelji. Baza podataka kao usluga je jedna od najviše korišćenih SaaS modela. Važno je napomenuti da je kod DBaaS modela korisnik usluga programer koji koristi usluge baze podataka za potrebe svoje aplikacije, za razliku od striming platformi koje predstavljaju takođe tip SaaS modela, ali korisnik ne mora da im bilo kakvo znanje o razvoju aplikacija, već kao korisnik klauda usluga predstavlja krajnjeg korisnika sistema. Na slici 2.4. se može videti da je korisnik DBaaS usluga aplikacija koju razvija programer ili skup programera.

  
Slika 2.4. Način funkcionisanja DBaaS [5]

Ideja koja stoji iza DBaaS modela je da programer konfiguriše i koristi bazu podataka preko API-ja (engl. *Application Programming Interface*) koji mu obezbeđuje klaud provajder. Preko ovog API-ja se može kreirati šema baze podataka, kao i upravljati podacima u istoj. Uglavnom je ovaj API grafički kako bi olakšao upravljanje bazom i podacima u njoj.

Prednosti korišćenja baze podataka kao usluge su sledeće:

* Finansijska isplatljivost - postavljanje infrastrukture kod tradicionalnog pristupa (storidži i raznorazne memorije) može biti veoma skupo i dugo, gde postoji rizik da količina resursa koju je kompanija incijalno obezbedila bude premala ili prevelika. Kod DBaaS modela, organizacija ima neograničen skup resursa na korišćenje za potrebe baze podataka, pri čemu provajderu plaća naknadu na osnovu resursa koje troši.
* Skalabilnost - DBaaS model omogućava lako dodavanje dodatnog storidža ili memorije ukoliko se za time javi potreba. Takođe, analogno povećanju količine resursa, moguće je i smanjiti količinu resursa koje koristi baza podataka ukoliko se uoči da baza podataka nekim danima ne koristi sve resurse koji su joj dati na raspolaganje.
* Jednostavnije održavanje - kod tradicionalnog pristupa je neophodno da postoji administrativni tim baze podataka koji će upravljati njom. DBaaS model prepušta klaud provajderu upravljanje bazom podataka, odnosno kreiranjem rezervnih kopija kao i ažuriranjem baze podataka pri čemu programeri mogu više vremena posvetiti razvoju samih aplikacija.
* Dostupnost baze podataka - kod DBaaS modela klaud provajder mora voditi računa o tome da li je baza podataka dostupna ili ne. Ukoliko baza podataka nije dostupna duže od onoga što je predviđeno potpisanim SLA ugovorom, onda klaud projvader mora da plati odštetu korisniku.

Kao i kod svakog modela, gde ima prednosti ima i mana. Kada neki pristup ili model ne bi imao mana, onda ostali modeli ne bi postajali. Glavne mane korišćenja baze podataka kao usluge su:

* Privatnost podataka - kako korisitmo usluge baze podataka od strane klaud provajdera, samim tim se baza podataka a i sami podaci sa njom nalaze na njihovim serverima. S toga korisniik DBaaS modela nema apsolutnu kontrolu nad tim ko sve ima pristup podacima.
* Suverentitet nad podacima - za kompaniju koja koristi bazu podataka kao uslugu se ne može reći da ima suverenitet nad tim podacima. Na primer, zamislimo situaciju u kojoj srpska velika korporacija koristi bazu podataka kao uslugu od strane američkog klaud provajdera. U slučaju rata ove dve države ili pojave konflikta između ove srpske kompanije i njoj konkurentne američke kompanije, američki klaud provajder može bez ikakvog saznanja odati tajne podatke srpske kompanije američkoj kompaniji. Da stvar bude još gora, postoji mogućnost da klaud provajder u potpunosti onemogući pristup bazi podataka srpskoj kompaniji. Neki bi rekli da je srpska kompanija u toj situaciji zaštićena ugovorom, ali - ko još u ratu poštuje pravila?
* Lokacija podataka - vlade određenih država ne dozvoljavaju da se lični podaci njihovih građana nalaze van granica zemlje, tako da sistemi koji bi radili sa privatnim i ličnim podacima građana ne bi smeli da koriste DBaaS pristup za svoje potrebe.

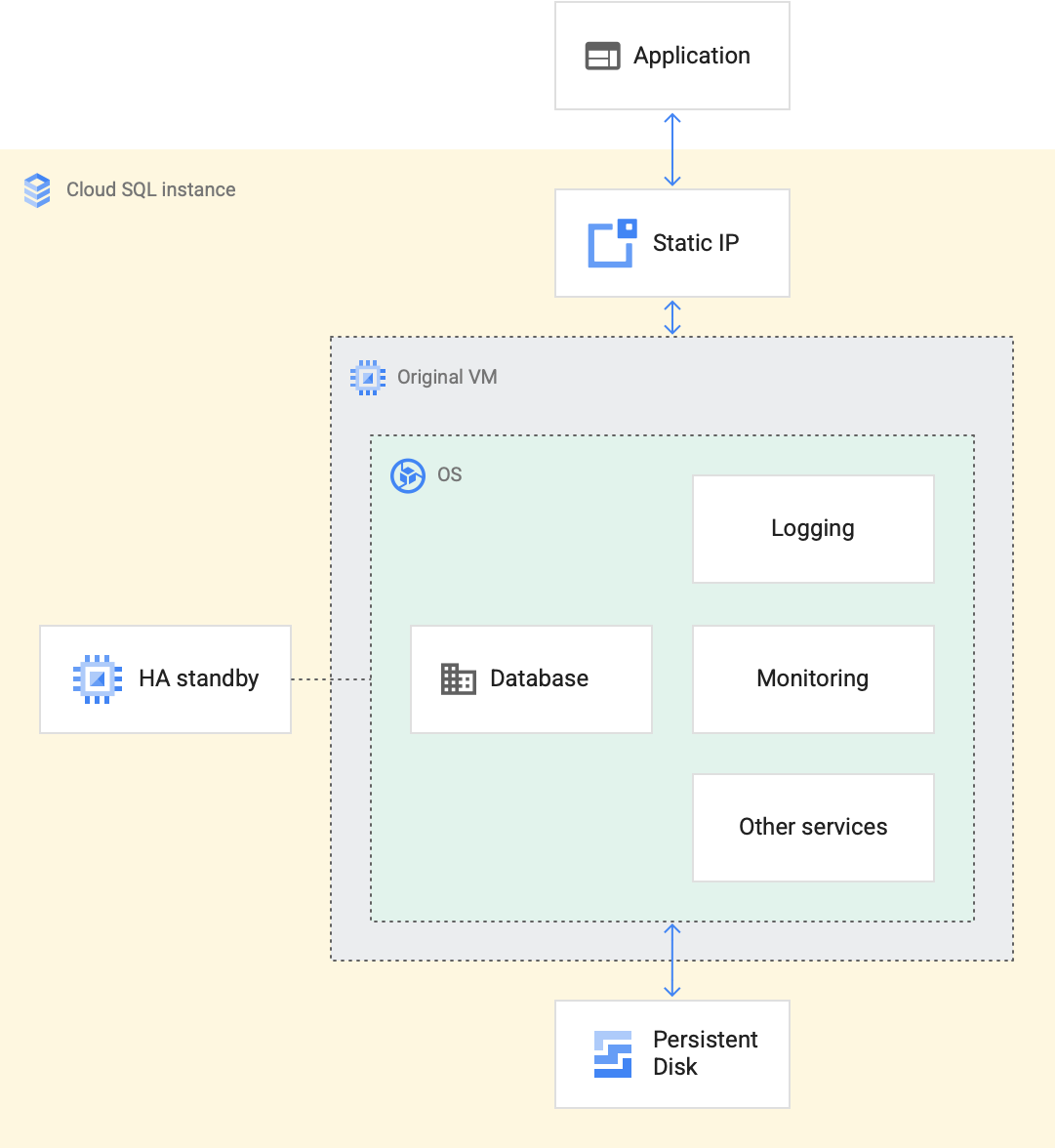
Najpoznatija DBaaS rešenja koja su na raspolaganju korisnicima su:

1. Amazon DynamoDB
2. Amazon Relational Database Service (RDS)
3. Google Cloud
4. MongoDB Atlas
5. Oracle Database
6. Azure SQL Database

Tema ovog seminarskog rada - Google Cloud SQL je jedan tip Google Cloud rešenja koje nudi podršku za korišćenje relacionih baza podataka. Pored Google Cloud SQL-a, postoji i skup NoSQL, In-Memory i Data Warehouse DBaaS rešenja koje nudi Google Cloud.

# Google Cloud SQL

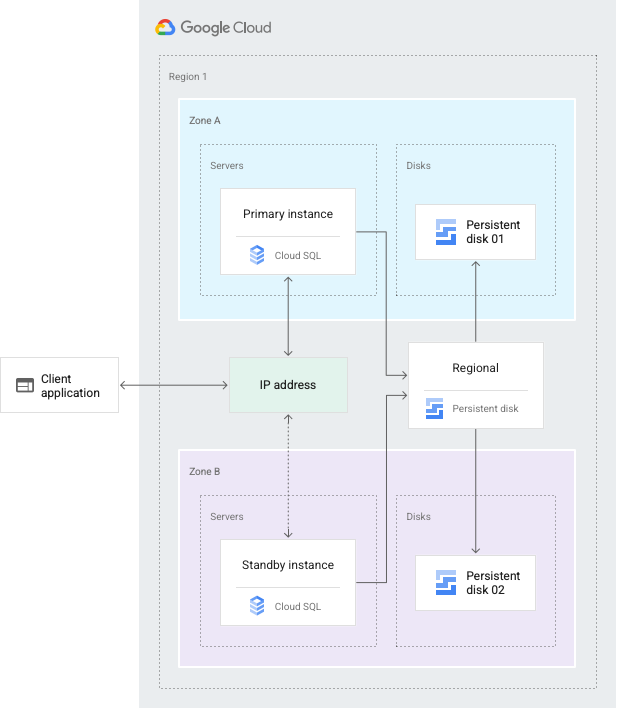
**Google Cloud SQL** [6] je DBaaS rešenje za relacione baze podataka od kompanije Google. Klijent može koristiti MySQL, PostgreSQL ili SQL Server u okviru ovog DBaaS rešenja, tako da predstavlja alternativu za korišćenje ovih baza podataka na lokalnom serveru. Veliki broj aplikacija koje se pokreću na *Google App Engine*-u, a isto tako veliki broj servisa koji nudi Google Cloud koristi upravo Cloud SQL za perzistenciju podataka.

  
Slika 3.1. Arhitektura Google Cloud SQL rešenja [6]

Arhitektura Google Cloud SQL DBaaS rešenja je prikazana na slici 3.1. Svaka instanca Cloud SQL-a je pokrenuta na virtuelnoj mašini koja se nalazi na Google Cloud-u. Unutar virtuelne mašine je pokrenut DBMS (engl. *Database Management System*) koji je zahtevao korisnik, kao i ostali servisi, od kojih neki pružaju podršku za logovanje događaja i nadgledanje stanja instance. Podaci same baze su smešteni na storidžu. Svaka VM sadrži statičku IP adresu (koja može biti javna ili privatna) kako bi klijent mogao da u svakom momentu pristupi instanci Cloud SQL-a.

U slučaju odabira neke baze podataka koju nudi Google Cloud SQL, klijent ne mora brinuti o sledećim stvarima s obzirom na to da se one podrazumevaju:

* Kreiranje rezervnih kopija (engl. *backups*) - bekapovi koje kreira Google Cloud SQL su inkrementalne prirode, tako da sadrže samo podatke koji su modifikovani/dodati nakon poslednjeg bekapa. Bekapovi se mogu kreirati na zahtev klijenta ili automatski, pri čemu se kopije prave svakodnevno. Lokacija novokreiranog bekapa može biti podrazumevana, pri čemu Cloud SQL određuje gde će se kopija čuvati, a postoji i mogućnost da klijent odabere gde želi da čuva bekapovi svoje baze.
* Visoka dostupnost i oporavak prilikom otkaza (engl. *High Availability*) - visoku dostupnost u Google Cloud SQL-u je moguće postići zbog postojanja redudantnosti među podacima. U tom slučaju, baza podataka ima svoju primarnu i rezervnu (engl. *standby*) instancu, gde primarna instanca predstavlja operativnu instancu (Slika 3.2). Rezervna instanca je pokrenuta na različitoj virtuelnoj mašini, na različitom fizičkom serveru od primarne instance, sa konfiguracijom koja je identična primarnoj instanci. Prilikom izvršenja transakcije nad primarnom instancom, rezervna instanca takođe ažurira svoje podatke kako bi oni bili konzistentni sa primarnom instancom. U slučaju pojave otkaza primarne instance, rezervna instanca postaje primarna instanca i ostaje primarna čak i u momentu oporavka prethodne primarne instance [7].

  
Slika 3.2. Način postizanja visoke dostupnosti Google Cloud SQL rešenja [7]

* Povezivanje na bazu podataka - pod ovim pojmom se podrazumevaju mogući načini povezivanja na kreiranu instancu, tj. da li klijent želi da instanca bude javno dostupna ili dostupna isključivo preko VPN-a, da li korisnik želi da koristi autentifikaciju i autorizaciju, da li saobraćaj ka i od baze podataka treba biti enkriptovan ili ne, itd...
* Uvoz i izvoz podataka (engl. *import and export*)
* Održavanje softvera baze podataka, odnosno DBMS-a i njegovo ažuriranje
* Monitoring (srp. *nadgledanje*) instance - instanca se može nadgledati pomoću Cloud Monitoring kontrolne table (engl. *dashboard*), što podrazumeva praćenje stanja baze podataka. Klijentu je takođe omogućeno podešavanje upozorenja - npr. klijent može podesiti da mu upozorenje u vidu poruke stigne na *mail* adresu kada zauzetost memorije Cloud SQL instance pređe prag od 75%.
* Logovanje događaja koji su se desili na instanci

Iako je navedeno u osobinama Google Cloud SQL-a da se provajder brine o ažuriranjima i izmenama, to ne mora da bude uvek slučaj. Ažuriranja instance se mogu podeliti u dve kategorije: **ažuriranje konfiguracije** i **ažuriranje sistema**.

Ažuriranje konfiguracije obavlja korisnik DBaaS usluge. Korisnik može promeniti konfiguraciju svoje baze podataka ukoliko se za tim javi potreba, a neke od tih situacija su npr. povećanje resursa zbog povećanja klijentskih zahteva i omogućenje visoke dostupnosti baze. Za promenu nekih konfiguracija nije neophodno da se restartuje ceo sistem, dok ažuriranje nekih konfiguracija zahteva određeni *downtime* (srp. *nedostupnost*) sistema. Zbog toga je najbolje da se ovakva ažuriranja izvršavaju u vremenskom periodu kada klijenti najmanje koriste ili uopšte ne koriste sistem.

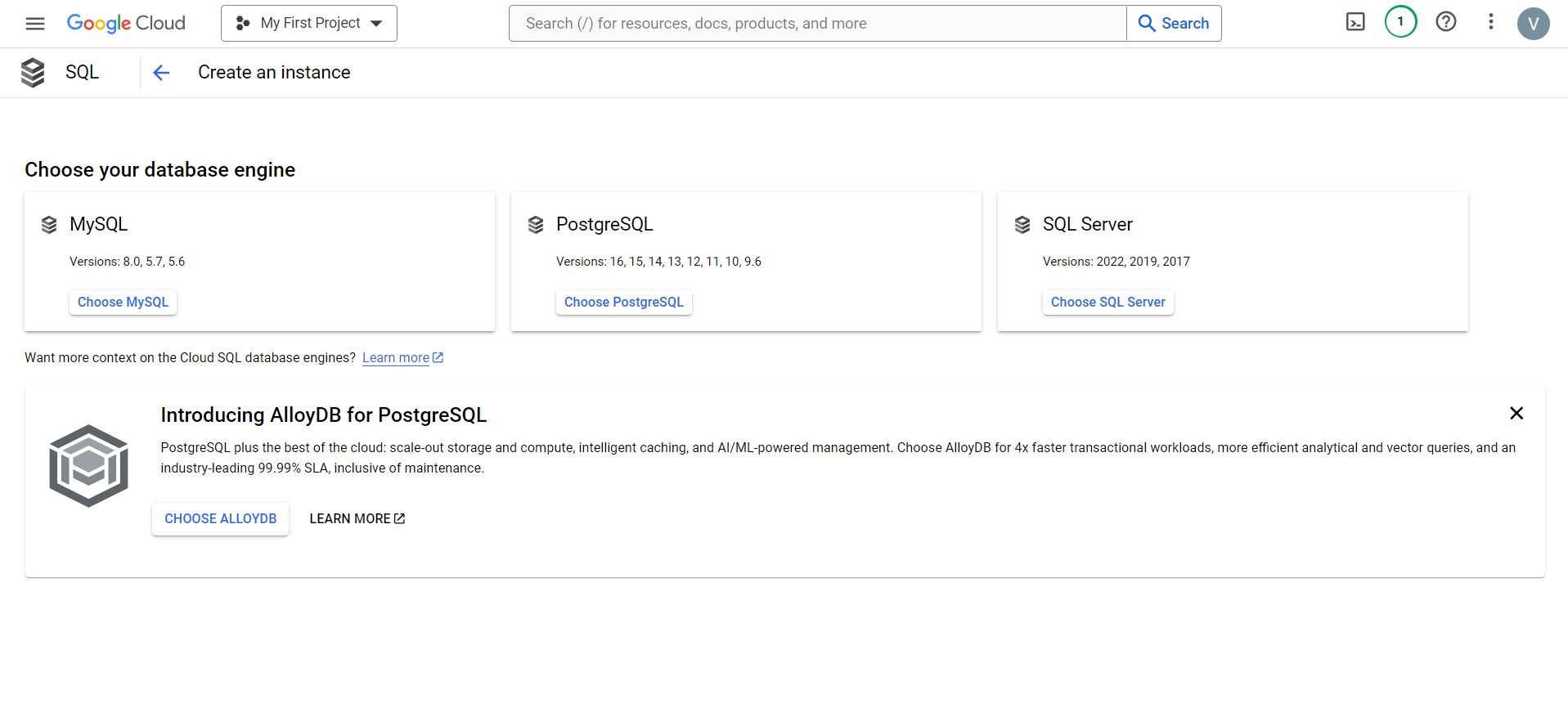
Ažuriranje sistema odvija sam klaud provajder, odnosno u ovom slučaju Cloud SQL. Ova ažuriranja se mogu podeliti u 3 osnovne kategorije:

1. Hardverska ažuriranja - pod ovim tipom ažuriranja se podrazumeva zamena i nadogradnja hardevrskih resursa servera na kome se izvršava instanca korisnika, i storidža koji koristi instanca. Zahvaljujući migracijom virtuelnih mašina koja se dešava uživo, korisnici ne primećuju nikakav *downtime* prilikom ažuriranja hardevrskih resursa. Migracija predstavlja prenos virtuelne mašine sa jednog servera na drugi server.
2. Onlajn ažuriranja - podrazumevaju ažuriranje pomoćnih servisa koji su pokrenuti na virtuelnoj mašini na kojoj se nalazi instanca korisnika. Ova ažuriranja se izvode dok je instanca pokrenuta, tako da *downtime* ne postoji.
3. Održavanje - pod ovim tipom spada ažuriranje operativnog sistema virtuelne mašine i samog softvera DBMS-a. Pošto ova ažuriranja zahtevaju restartovanje instance, klijenti aplikacije mogu primetiti *downtime*, ali Cloud SQL nudi mogućnost odabira vremena kada će se ažuriranje obaviti.

## Kreiranje instance Google Cloud SQL-a

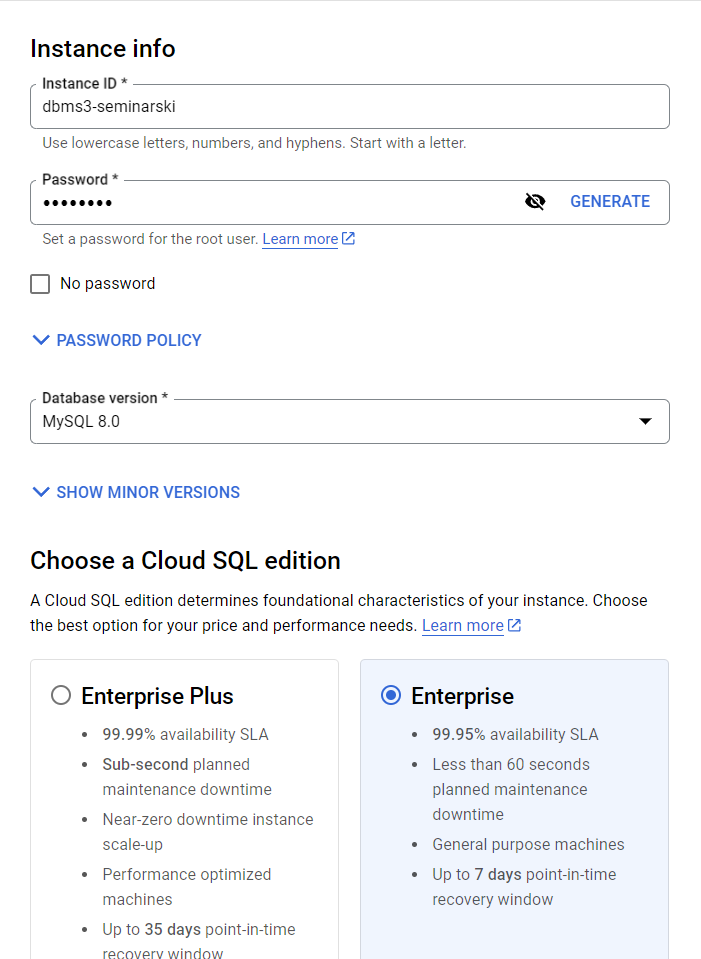
U ovom poglavlju će biti prikazano kreiranje instance Google Cloud SQL-a, konkretno MySQL instance.

Google Cloud novim korisnicima nudi kredit od 300$ da isprobaju Cloud SQL koji mogu potrošiti u roku od devedeset dana. Ipak, prilikom kreiranja instance, neophodno je ostaviti podatke svoje platne kartice kako bi Google Cloud smanjio prevare i zloupotrebe. Automatska naplata usluga nakon isteka probnog perioda od 90 dana neće krenuti sve dok korisnik samoinicijativno ne aktivira svoj Google Cloud nalog. Kod kreiranja instance, Google Cloud će ponuditi mogućnost odabira baze (Slika 3.3).

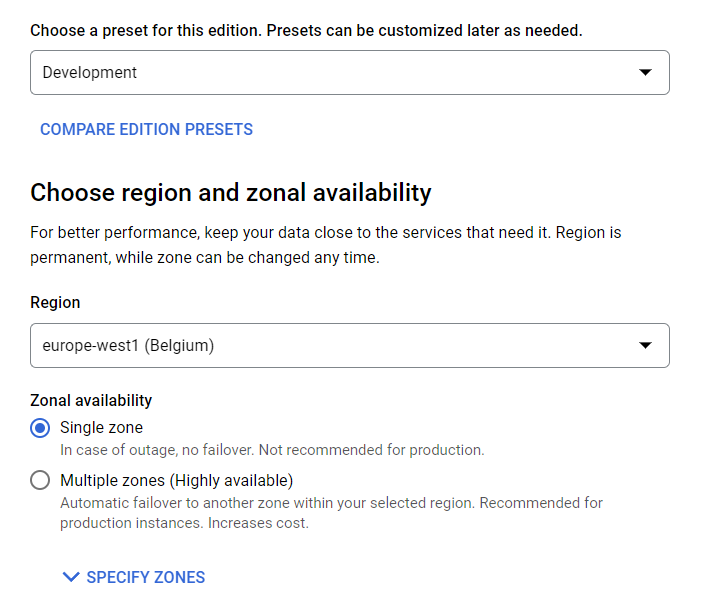
  
Slika 3.3. Prvi ekran prilikom kreiranja instance

Da bi kreiranje instance bilo moguće, neophodno je omogućiti *Compute Engine API*. Razlog tome je što se preko ovog API-ja kreira virtuelna mašina koja će hostovati instancu baze podataka. U jednom od prethodnih poglavlja je rečeno da je DBaaS rešenje jedan tip SaaS rešenja i da kao takav pruža najveći nivo apstrakcije korisniku za korišćenje usluga, ali je kod kreiranja instance baze podataka (SaaS pristup), kao i kod kreiranje virtuelne mašine (IaaS pristup), korisniku omogućen odabir količine hardverskih resursa koje želi da poseduje njegova instanca. Sada se posatavlja pitanje - kako SaaS, odnosno DBaaS pristup pruža najviši nivo apstrakcije ukoliko korisnik takođe mora da bira količinu resursa kao i kod IaaS pristupa. Razlika je u tome što kod SaaS pristupa korisnik bira sa kojom količinom resursa će raspolagati virtuelna mašina na kojoj je pokrenut uslužni softver, pri čemu on ne mora da se brine o održavanju same virtuelne mašine i softvera, dok kod IaaS pristupa korisnik bira resurse virtuelne mašine i sam se treba brinuti o operativnom sistemu i uslužnim softverima koji su pokrenuti na tom operativnom sistemu. Dakle, ključna razlika između različitih tipova usluga leži u delovima virtuelne mašine kojima korisnik može da upravlja.

Prilikom kreiranja instance, korisniku će biti ponuđen odabir količine resursa koje će koristiti instanca, ali pre toga je neophodno navesti *Instance ID*, lozinku, verziju željenog DBMS-a (u ovom slučaju MySQL-a) i koj paket Cloud SQL-a korisnik želi (u mogućnosti su *Enterprise Plus* i *Enterprise*), kao što je prikazano na slici 3.4.

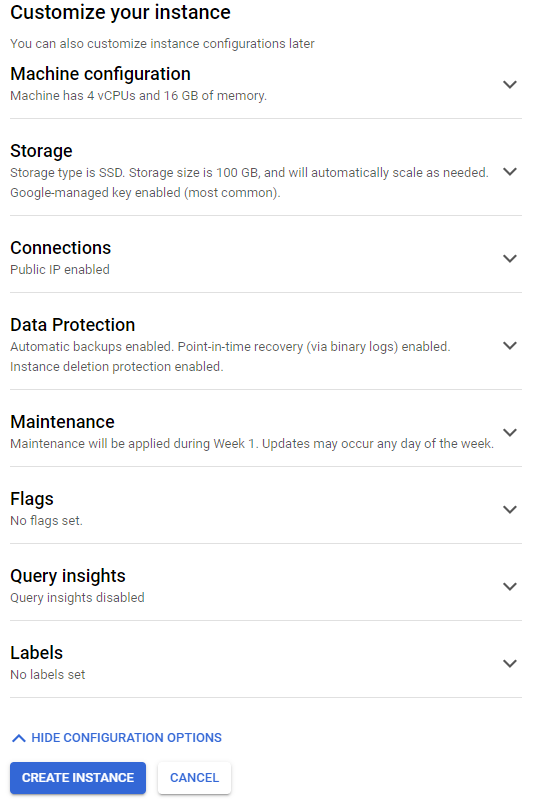
  
Slika 3.4. Postavljanje osnovnih podataka novokreirane instance

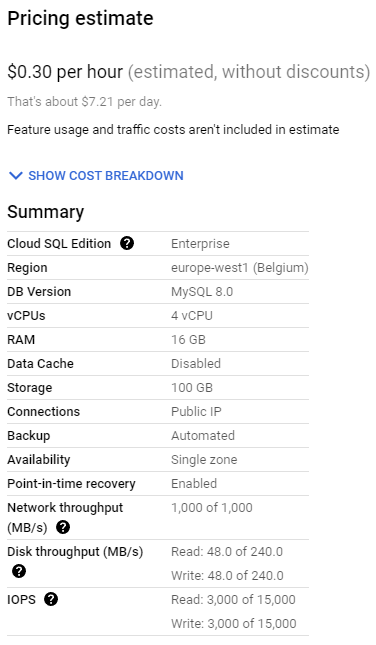
Posle ovoga je neophodno selektovati kako će baza biti korišćena - u produkciji ili u razvoju. Nakon toga će odabir regiona i zone biti zahtevan od korisnika. Kod *Zonal availability* opcije je moguće odlučiti da li želimo da koristimo High Availability rešenje ili ne (Slika 3.5).

  
Slika 3.5. Odabir namene baze podataka i regiona gde će se instanca nalaziti

Nakon navedenih opcija moguće je izmeniti podrazumevana podešavanja instance koja se tiču konfiguracija virtuelne mašine na kojoj će instanca biti pokrenuta (Slika 3.6). Ostavljena su podrazumevana podešavanja - mašina sa 4 procesora, 16GB RAM memorije, SSD storidž od 100GB, javno dostupna instanca (javna IP adresa), omogućen bekap i održavanje instance jednom nedeljno.

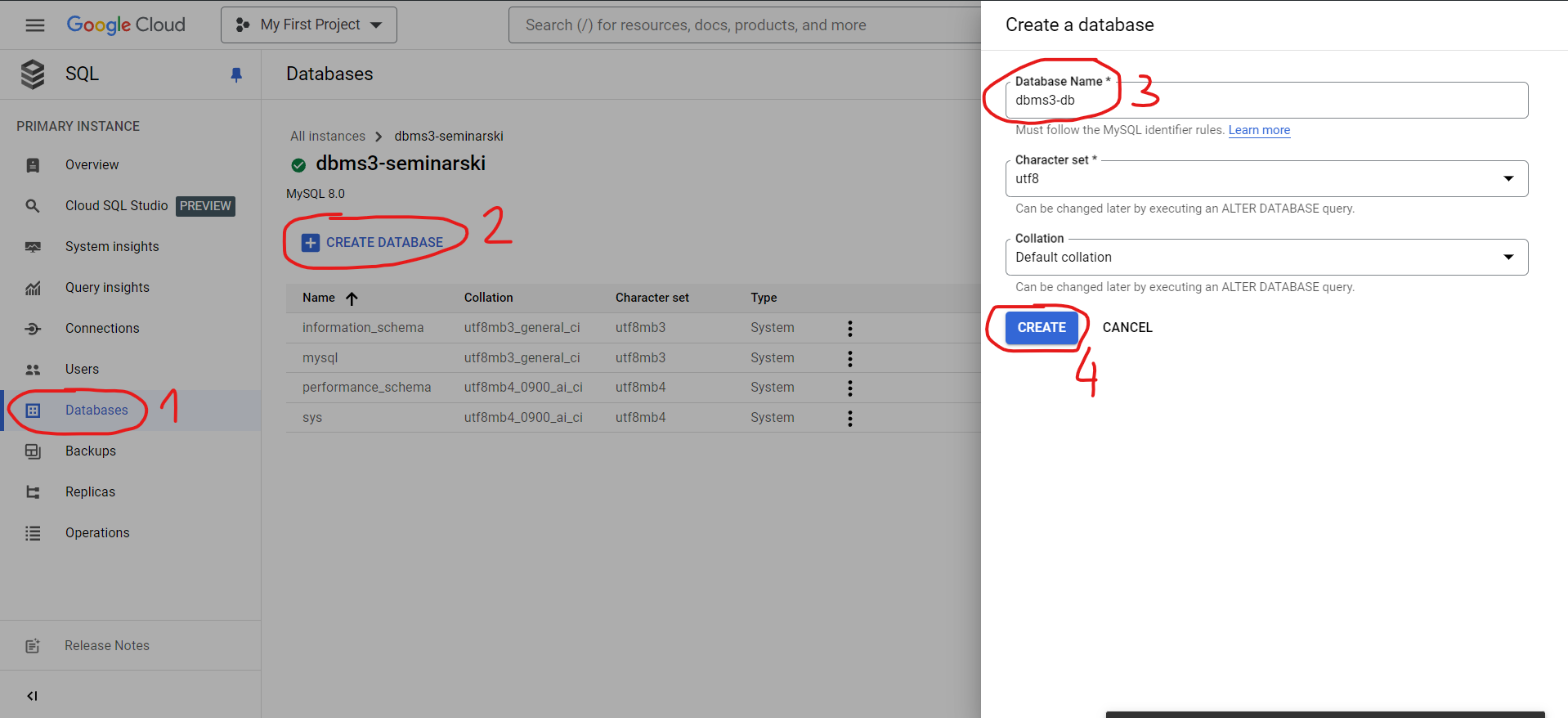
Prilikom odabira svih osobina i opcija instance, Cloud SQL će nam prikazati iznos održavanje jedne takve instance kao što je prikazano na slici 3.7. Kako bi se instanca kreirala, neophodno je kliknuti CREATE INSTANCE dugme koje je prikazano na slici 3.6.

  
Slika 3.6. Podrazumevana podešavanja virtuelne mašine na kojoj će biti pokrenuta instanca Cloud SQL-a

  
Slika 3.7. Naplata Cloud SQL instance sa prikazanim osobinama

## Kreiranje instance baze podataka

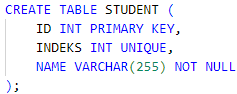
Nakon uspešnog kreiranja instance, neophodno je da se kreira baza podataka u kojoj će se kreirati relacije i podaci. Potrebno je otići na *Databases* karticu iz menija, zatim kliknuti CREATE DATABASE dugme i postaviti naziv baze podataka kao što je prikazano na slici 3.8.

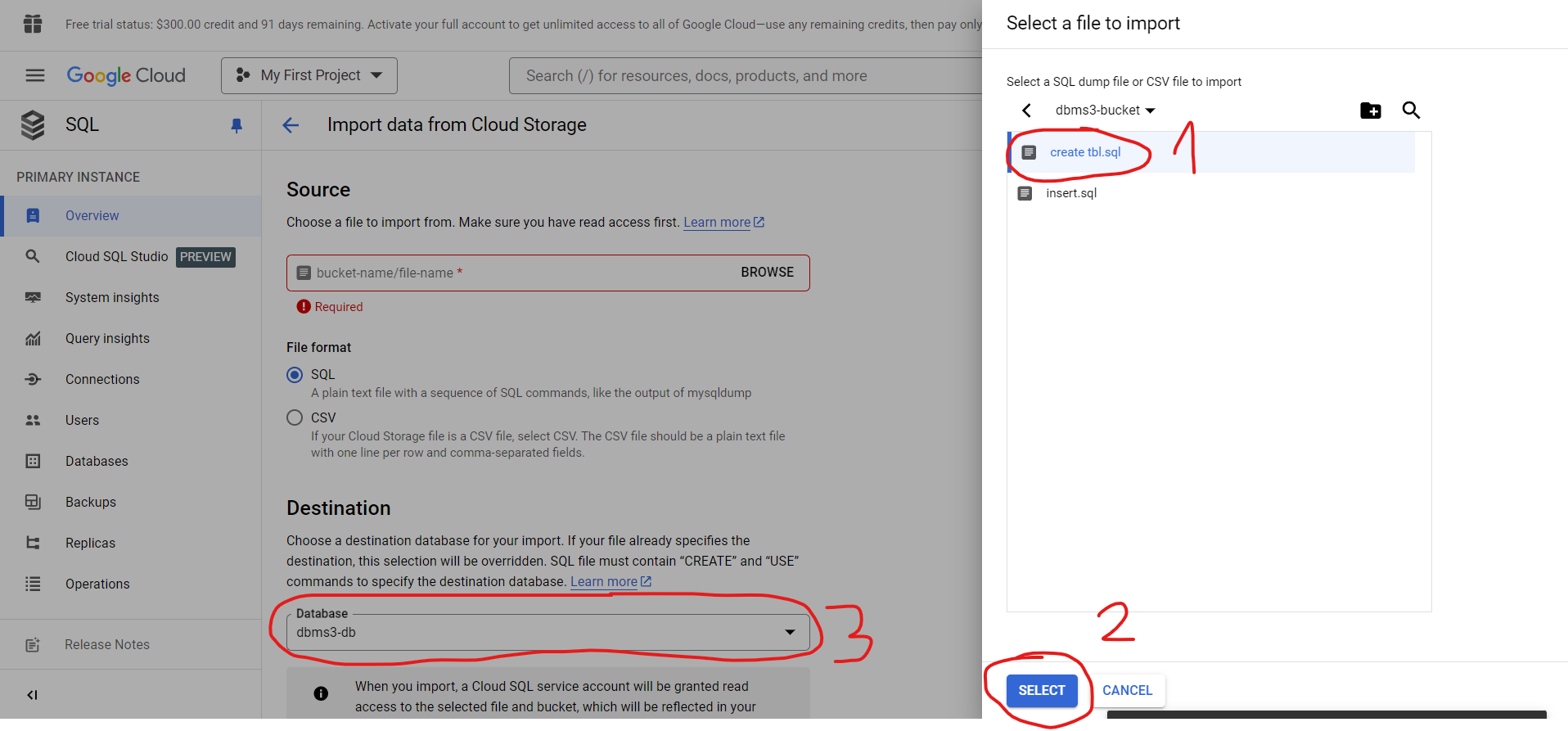
  
Slika 3.8. Kreiranje baze podataka

Rad sa bazom podataka je moguć na 2 načina:

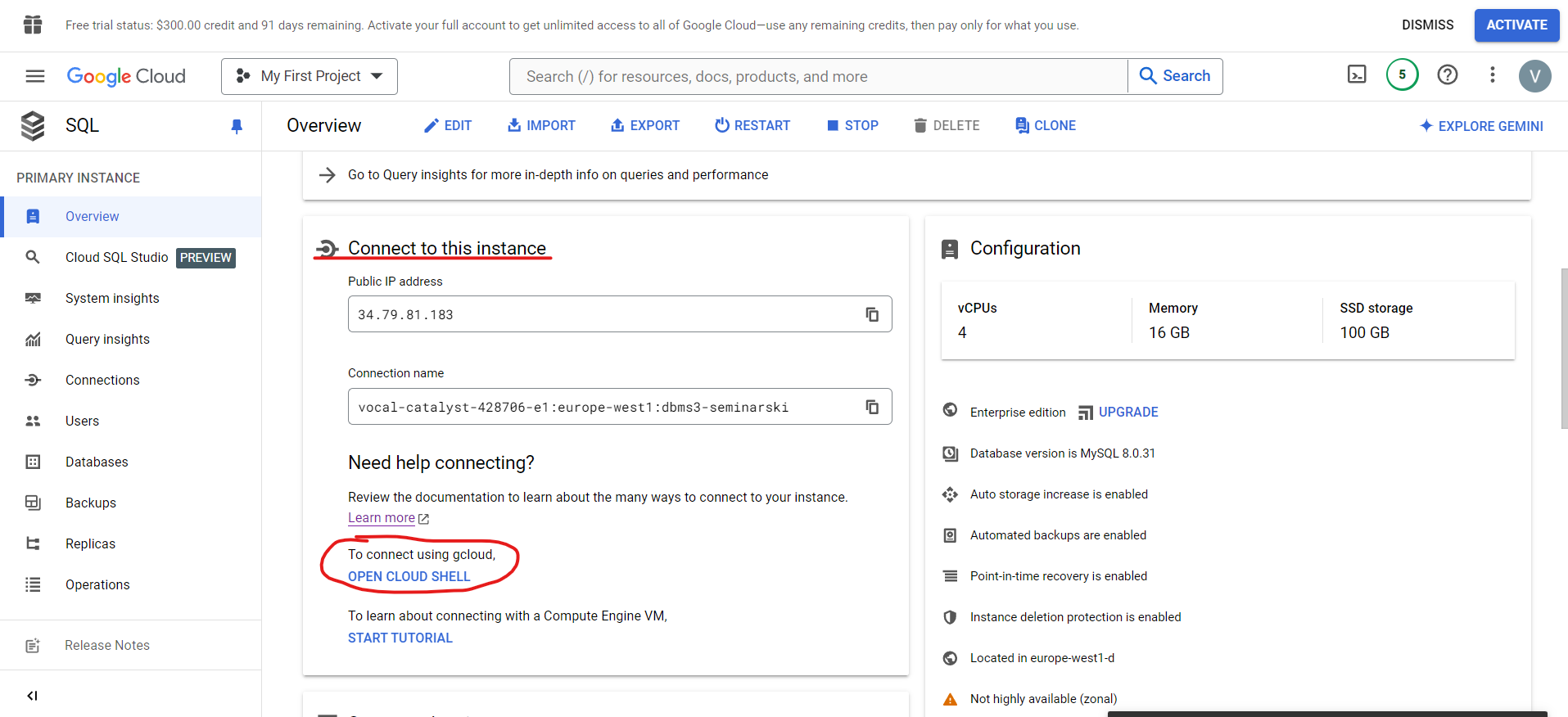
1. Importovanjem sql skripti koje će se izvršiti nad instancom baze podataka
2. Korišćenjem konzole baze podataka

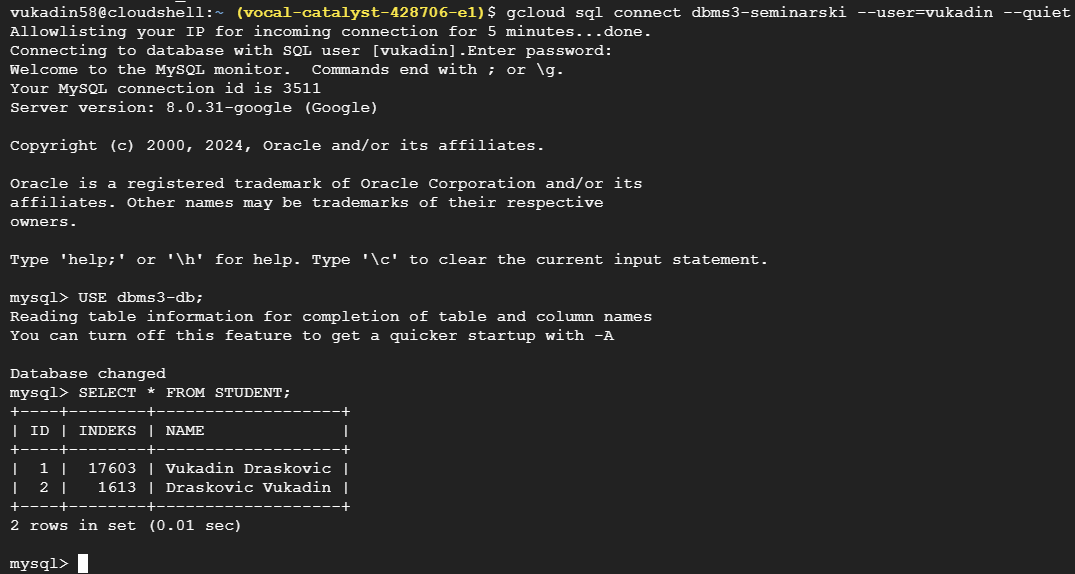
Napisana SQL skripta koja će kreirati relaciju STUDENT (Slika 3.9), biće importovana u *dbms3-db* bazu podataka kao što je prikazano na slici 3.10. Neophodno je da ova skripta bude smeštena na Google Cloud storidžu kako bi mogla biti izvršena.

  
Slika 3.9. Sadržaj CREATE TABLE skripte

  
Slika 3.10. Upload-ovanje sql skripte u bazu podataka

Na isti način kao što je upload-ovana SQL skripta za kreiranje relacije će biti dodata i izvršena SQL skripta za unos podataka u relaciju. Kako bi se uverili da su podaci uspešno uneti u bazu podataka, SELECT naredba će biti izvršena iz terminala. Kako bi se otvorio terminal, neophodno je iz *Connect to this instance* kontrolne table kliknuti OPEN CLOUD SHELL dugme (Slika 3.11). Da bi korišćenje terminala bilo moguće, neophodno je omogućiti *Cloud SQL Admin API*. Nakon toga će biti zahtevano od korisnika da unese lozinku kako bi mogao da koristi konzolu. Na slici 3.12. je prikazan rezultat izvršenja SELECT naredbe.

  
Slika 3.11. Otvaranje konzole kreirane instance

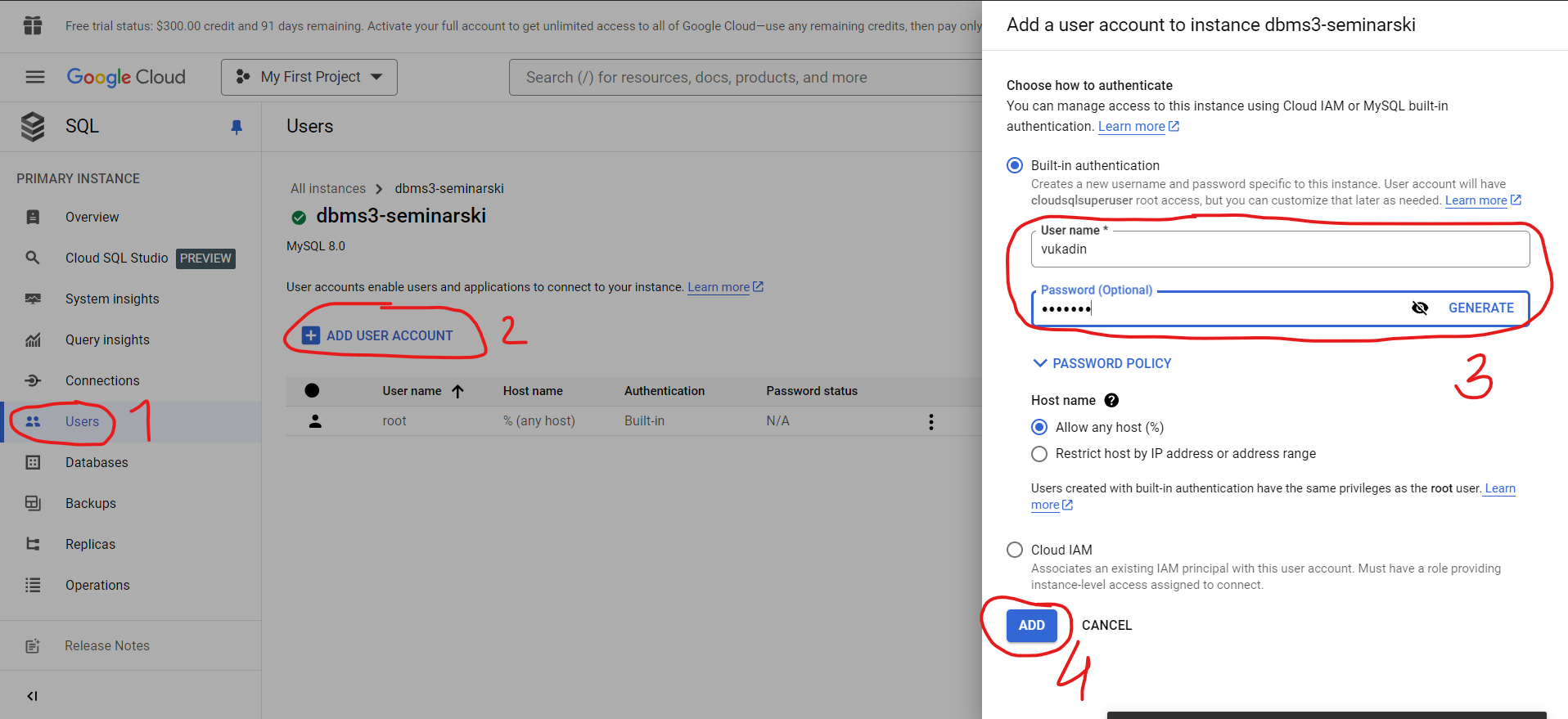
  
Slika 3.12. Korišćenje konzole

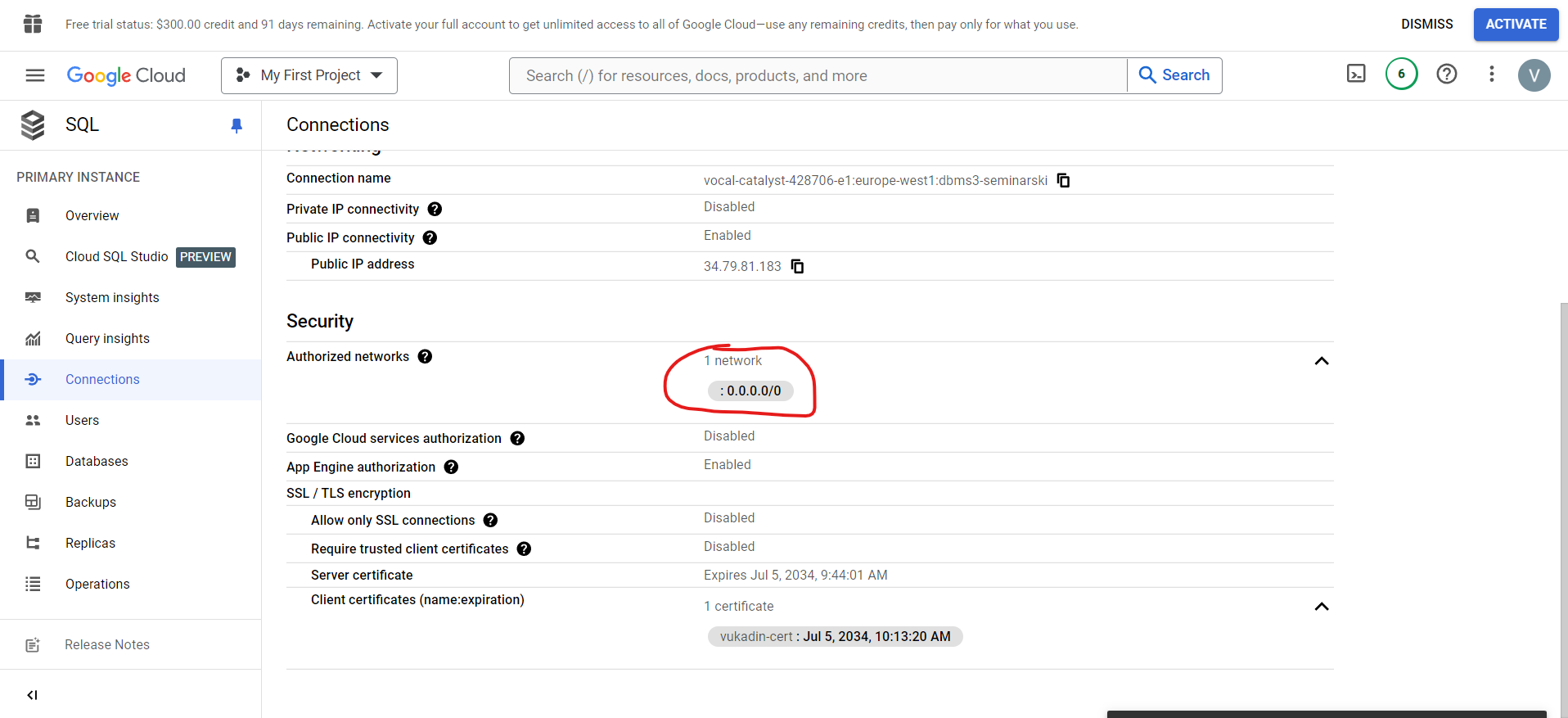
## Povezivanje aplikacije na Cloud SQL instancu

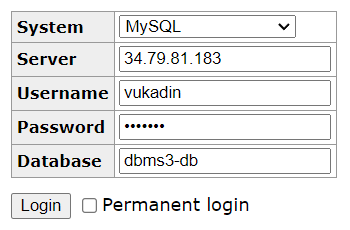
U ovom poglavlju će biti opisan postupak povezivanja aplikacije na novokreiranu Cloud SQL instancu. Za ove potrebe je iskorišćen *Adminer* program [8] kao simulacija aplikacije koja koristi bazu podataka. Adminer se pokreće kao *Docker kontejner* [9].

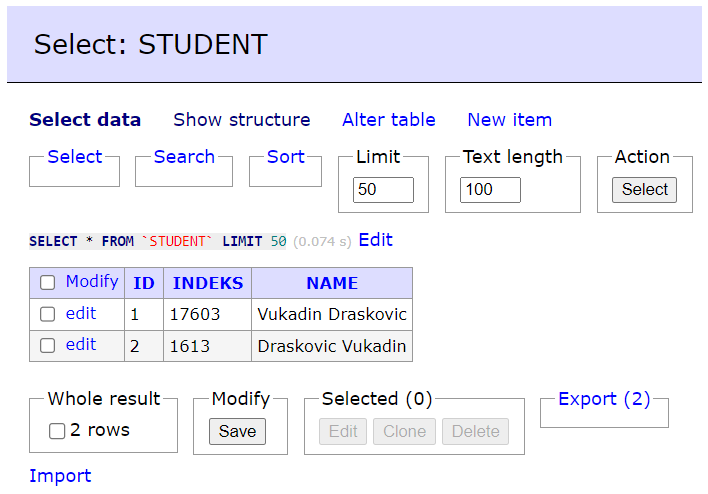
Na početku je neophodno kreirati korisnika kojim će se aplikacija logovati na bazu podataka (Slika 3.13). Nakon toga je neophodno omogućiti sa kojih IP adresa instanca prihvata konekcije. U ovom slučaju je omogućeno da instanca prihvata konekcije sa bilo koje IP adrese, tako da je iz tog razloga postavljena adresa 0.0.0.0 sa maskom 0 (Slika 3.14). U stvarnom primeru bi bazu podataka koristio servis koji ima javnu statičku IP adresu, tako da bi tu mogla biti navedena upravo ta IP adresa na kome se hostuje servis koji koristi bazu podataka.

Sada je potrebno povezati aplikaciju na bazu podataka. U delu za unos kredencijala je neophodno navesti podatke kao na slici 3.15. Na slici 3.16. je prikazan izgled *Adminer* aplikacije nakon uspešnog logovanja i izvršenja SELECT naredbe nad STUDENT relacijom u *dbms3-db* bazi podataka.

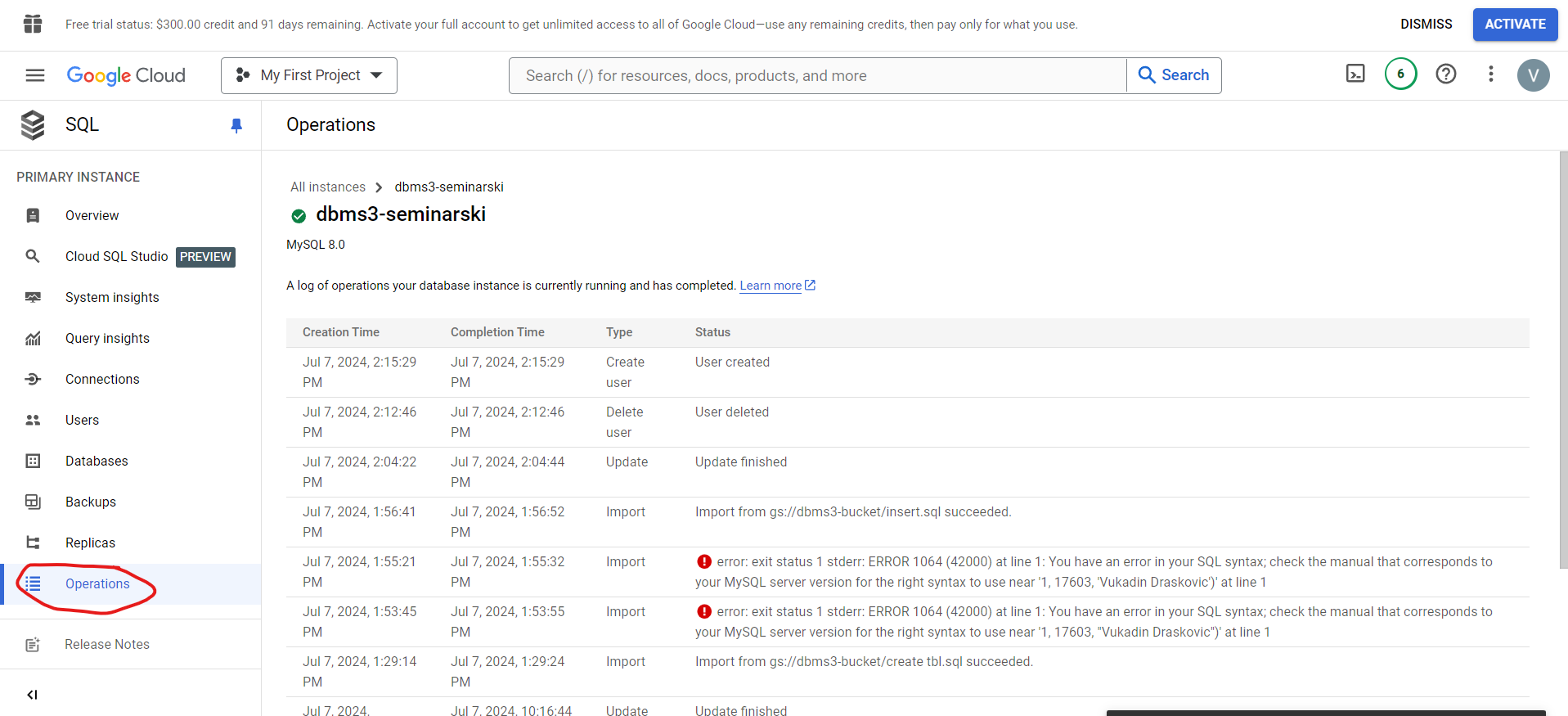
  
Slika 3.13. Kreiranje korisnika za potrebe aplikacije

  
Slika 3.14. Prihvatanje konekcija sa određenih IP adresa

  
Slika 3.15. Povezivanje na Cloud SQL instancu iz *Adminer* aplikacije

  
Slika 3.16. Interfejs *Adminer* aplikacije

Za kraj, na slici 3.17. je prikazan izgled logova koje generiše instanca baze podataka.

  
Slika 3.17. Logovi instance baze podataka

# Zaključak

U radu su prvo predstavljeni osnovni koncepti i pojmovi računarstva u oblaku koji predstavlja ključni koncept za korišćenje DBaaS rešenja. Nakon toga je dat spisak karakteristika DBaaS rešenja, njegovih prednosti i mana. Na primeru Google Cloud SQL-a je prikazan način korišćenja jednog DBaaS rešenja, gde je svaki korak upotrebe ilustrovan slikom ekrana. Za simulaciju aplikacije koja koristi DBaaS rešenje je iskorišćen Adminer program, gde se, naravno, umesto tog programa može naći aplikacija koju razvija tim programera.

Cloud baze podataka i DBaaS rešenja značajno pojednostavljuju korišćenje baze podataka. Korisnik ne mora da vodi računama o verziji DBMS-a, njegovoj instalaciji, hardveru na kome je DBMS pokrenut, logovima i ostalim stvarima navedenim u radu. Međutim, sve ima svoju cenu. Privatnost klijenata i korisnika se dovodi u pitanje, kao i sigurnost i bezbednost njihovih podataka. Zaključak je sledeći - DBaaS rešenje je savršeno za aplikacije i sisteme koje ne rade sa podacima od visoke važnosti. Ukoliko podaci trebaju ostati tajni i privatni za bilo koga osim za korisnika sistema, korišćenje Cloud baza podataka i DBaaS rešenja se ne preporučuje.

# Literatura

[1] Rajkumar Buyya, James Broberg, Andrzej Goscinski - CLOUD COMPUTING Principles and Paradigms, WILEY

[2] Type 1 and Type 2 Hypervisors: What Makes Them Different, dostupno na: <https://medium.com/teamresellerclub/type-1-and-type-2-hypervisors-what-makes-them-different-6a1755d6ae2c> (pristupljeno 2. jula 2024.)

[3] What is a DBaaS?, dostupno na: <https://severalnines.com/blog/what-is-a-dbaas/> (pristupljeno 3. jula 2024.)

[4] What is database as a service (DBaaS)?, dostupno na: <https://www.ibm.com/topics/dbaas> (pristupljeno 4. jula 2024.)

[5] Medical imaging services supported on cloud, dostupno na: <https://www.researchgate.net/publication/260374716_Medical_imaging_services_supported_on_cloud> (pristpuljeno 4. jula 2024.)

[6] Cloud SQL, dostupno na: [https://cloud.google.com/sql?hl=en#focus-on-your-application-and-leave-the-database-to-us](https://cloud.google.com/sql?hl=en%23focus-on-your-application-and-leave-the-database-to-us) (pristpuljeno 5. jula 2024.)

[7] About high availability, dostupno na: <https://cloud.google.com/sql/docs/mysql/high-availability> (pristpuljeno 7. jula 2024.)

[8] Adminer, dostupno na: <https://www.adminer.org/> (pristupljeno 7. jula 2024.)

[9] adminer docker image, dostupno na: <https://hub.docker.com/_/adminer/> (pristupljeno 7. jula 2024.)