



**Univerzitet u Nišu, Elektronski fakultet
Katedra za računarstvo**



Cloud SQL kao database-as-a-service rešenje

Sistemi za upravljanje bazama podataka

Seminarski rad

Mentor:

Doc. dr Aleksandar Stanimirović

Student:

Vladana Stojiljković 1135

Niš, 2021.

Sadržaj

1.	Uvod.....	1
2.	Cloud SQL	2
2.1.	Funkcionalnosti Cloud SQL-a	3
2.1.1.	Rezervne kopije	4
2.1.2.	Replikacija	5
2.1.3.	Visoka dostupnost i <i>failover</i>	7
2.1.4.	Oporavak od katastrofe	10
3.	MySQL na Cloud SQL-u	14
3.1.	Kreiranje instance.....	14
3.2.	Povezivanje na instancu	16
3.3.	Kreiranje rezervnih kopija.....	18
3.4.	Kreiranje replika za čitanje	19
3.5.	HA konfiguracija.....	20
3.6.	Simulacija DR procesa	21
4.	Zaključak.....	24
5.	Literatura.....	25

1. Uvod

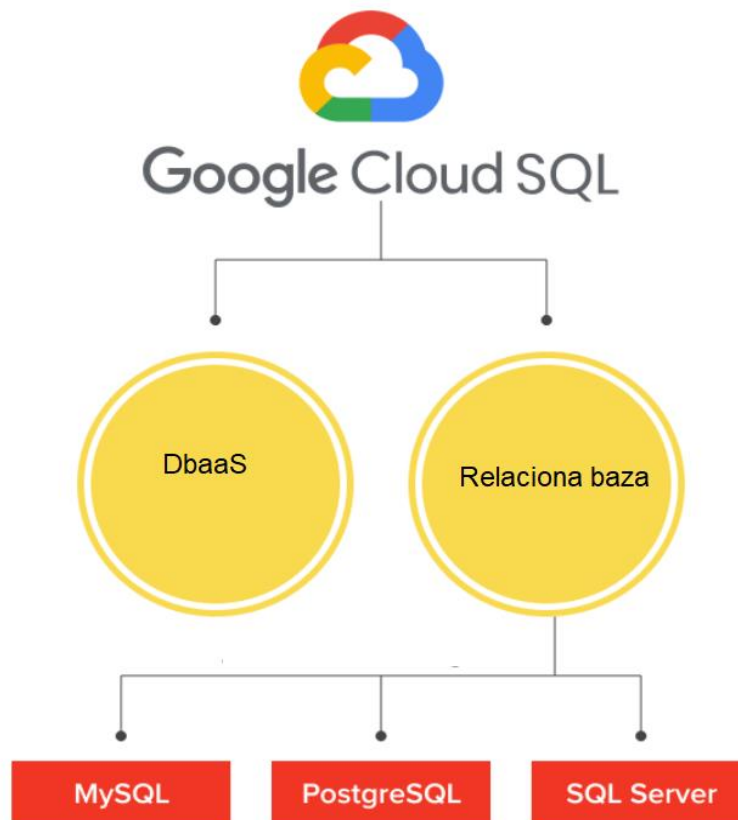
Database-as-a-service, skraćeno *DBaaS* (takođe poznat i kao usluga upravljane baze podataka) je usluga računarstva u oblaku koja omogućava korisnicima pristup i upotrebu sistema baze podataka u oblaku bez kupovine i postavljanja sopstvenog hardvera, instaliranja sopstvenog softvera baze podataka ili samog upravljanja bazom podataka. Dobavljač usluga u oblaku (eng. *cloud provider*) brine o svemu, od periodičnih nadogradnji i sigurnosnih kopija do osiguranja da sistem baza podataka ostane dostupan i siguran 24/7. Kod DBaaS korisnik može da podešava, upravlja i skalira baze podataka koristeći zajednički skup apstrakcija (primitiva), bez potrebe da zna ili brine o tačnim primenama tih apstrakcija za konkretnu bazu podataka [1].

Cloud SQL je potpuno upravljana usluga baze podataka koja olakšava postavljanje, održavanje i upravljanje PostgreSQL, MSSQL Server i MySQL relacionim bazama podataka u oblaku. Pošto je potpuno upravljana, provajder (Google Cloud) upravlja u celosti operacijama poput replikacije i oporavka, a ne korisnik. Cloud SQL nudi visoke performanse, skalabilnost i praktičnost. Hostovan na Google Cloud platformi, Cloud SQL pruža infrastrukturu baze podataka za aplikacije koje rade bilo gde. Cloud SQL upravlja instancom baze podataka, a korisnik samim podacima. Više reči o njemu biće u poglavlju 2, dok će konkretni primeri korišćenja Cloud SQL-a za MySQL bazu podataka biti prikazani u poglavlju 3. Zaključak je dat u poglavlju 4, a spisak korišćene literature u poglavlju 5.

2. Cloud SQL

Cloud SQL je potpuno upravljana (eng. *fully managed*), relaciona Google Cloud usluga baze podataka koja je kompatibilna sa SQL Server-om, MySQL-om i PostgreSQL-om (slika 1). Kao DbaaS rešenje, značajno olakšava upravljanje bazom podataka. Sadrži funkcije za automatizovane rezervne kopije, replikaciju podataka i oporavak od katastrofe kako bi se osigurala visoka dostupnost i elastičnost. Može da se integriše sa Compute Engine, App Engine, BigQuery i Kubernetes sistemom. Obično se koristi u sledećim situacijama [2]:

- za podizanje i premeštanje privatnih (eng. *on-premise*) SQL baza podataka u oblak
- za vršenje masovne analitike nad podacima
- kao podrška skadištenju i skalabilnosti sistema za upravljanje sadržajem (eng. *Content Management System, CMS*)
- za upravljanje bazama podataka korišćenjem infrastrukture kao koda (eng. *Infrastructure-as-a-Code, IaC*)
- za razvoj kontejnerizovanih aplikacija i mikroservisa.



Slika 1: Google Cloud SQL

Jedan od osnovnih pojmova u Cloud SQL-u je Cloud SQL instanca. Onda odgovara virtualnoj mašini koja sadrži instancu baze podataka i odgovarajući softver za kontejnerizaciju koji omogućava podizanje te baze [3]. Instanca baze podataka je skup softvera i datoteka koje upravljaju bazom podataka (MySQL, PostgreSQL ili SQLServer). Cloud SQL-om se upravlja iz *gcloud* komandne linije i komande su u obliku

gcloud sql GROUP / COMMAND [GLOUD_WIDE_FLAG ...].

Pomoću njih vrši se kreiranje i upravljanje bazom podataka. Vrednost za *GROUP* određuje koje komande mogu da se koristi i može da bude *databases*, *replicas*, *backups* itd. a komanda *COMMAND* može da bude *connect*, za povezivanje na Cloud SQL instancu, ili *reschedule-maintenance*, za određivanje rasporeda održavanja instance [4]. Na slici 2 dat je primer *gcloud* naredbe za prikaz svih baza podataka u sistemu. Skup naredbi koji pruža *gcloud* omogućava korisnicima da upravljaju bilo kojom od tri mogućih tipa baze podataka, što znači da ove naredbe apstrahuju njihove funkcionalnosti, pa korisniku nije neophodno da zna komande za konkretnu bazu (npr. MySQL).

```
eonrokkrl123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql databases list -i mysql1135
NAME          CHARSET  COLLATION
mysql         utf8     utf8_general_ci
information_schema utf8     utf8_general_ci
performance_schema utf8mb4  utf8mb4_0900_ai_ci
sys           utf8mb4  utf8mb4_0900_ai_ci
```

Slika 2: Primer *gcloud* naredbe

2.1. Funkcionalnosti Cloud SQL-a

Cloud SQL je DbaaS, pa omogućava različite usluge koje olakšavaju rad korisniku, tako da on može da se fokusira više na podatke, nego na samo upravljanje bazom podataka. U podržane usluge spadaju:

1. rezervne kopije (eng. *backup*)
2. visoka dostupnost i *failover*
3. eksportovanje i importovanje podataka
4. održavanje i ažuriranje
5. nadgledanje (eng. *monitoring*)
6. logovi.

Kad se koristi za MySQL ili PostgreSQL, pored ovih usluga moguća je i enkripcija podataka i replikacija.

Cloud SQL obezbeđuje i sigurnost jer se podaci šifruju kada se prenose Google-ovim unutrašnjim mrežama i kada se čuvaju u tabelama baza podataka, privremenim datotekama i rezervnim kopijama. Cloud SQL podržava privatno povezivanje putem virtuelnog privatnog oblaka (eng. *Virtual Private Cloud, VPC*), a svaka instanca Cloud SQL-a uključuje i mrežnu zaštitnu barijeru (eng. *firewall*), što omogućava korisnicima da kontrolišu pristup instanci baze podataka preko javnih mreža.

Još jedna prednost korišćenja Cloud SQL-a je skalabilnost jer je moguće dodati do 96 procesorskih jezgara, 624GB RAM memorije i 30GB za skladište [5]. Veličina skladišta može da se povećava automatski jer Cloud SQL vrši provere na određen vremenski period i proverava da li je slobodan prostor na izmaku. U slučaju da jeste, povećava ga, pa korisnik ne mora unapred da zna veličinu skladišta koje mu je potrebno.

U daljem tekstu biće pojašnjene neke od značajnijih karakteristika Cloud SQL-a.

2.1.1. Rezervne kopije

Rezervne kopije omogućavaju vraćanje izgubljenih podataka u CloudSQL instancu. Pored toga, ako na instanci dođe do greške, moguće je vratiti je u prethodno stanje pomoću rezervne kopije. Postoji dva tipa rezervnih kopija: automatske i na zahtev (eng. *on demand backups*) [6].

Automatske rezervne kopije se kreiraju korišćenjem 4-časovnih prozora, tokom kojih počinje njihovo pravljenje. Ti prozori u stvari predstavljaju vremenske periode u trajanju od 4 sata za vreme kojih se vrši pravljenje kopije. Preporuka je da se kopije kreiraju kad baza podataka ima najmanje aktivnosti. Automatske rezervne kopije se kreiraju svaki dan kada je pokrenuta instanca, ali samo za vreme prozora. Nakon zaustavljanja instance pravi se jedna dodatna automatizovana sigurnosna kopija kako bi se zaštitile sve promene pre zaustavljanja instance [6]. Podrazumevano se zadržava do sedam najnovijih rezervnih kopija. Maksimalan broj automatskih rezervnih kopija je konfigurabilan, od 1 do 365. Kreiranje automatskih rezervnih kopija se zaustavlja ako je instanca zaustavljena duže od 36 sati. Google Cloud SQL pravi rezervnu kopiju instance samo ako su se podaci instance promenili od poslednje rezervne kopije. U zavisnosti od vremena mirovanja instance između rezervnih kopija, vreme zadržavanja rezervne kopije može biti duže od 7 dana. Kada se izvrši vraćanje u prethodne stanje iz rezervne kopije, menjaju se svi podaci na instanci. Vrednosti zadržavanja evidencije rezervnih kopija i transakcija takođe mogu da se konfigurišu. U bilo kom trenutku moguće je omogućiti/onemogućiti pravljenje automatskih rezervnih kopija za instancu (npr. tokom neke dugotrajne operacije, poput učitavanja velike količine podataka).

Za potrebe kreiranja kopija kreira se novi korisnik *cloudsqladmin* za svaku instancu i generiše se lozinka za njega. Kad CloudSQL pravi kopije, loguje se kao *cloudsqladmin*. Rezervne kopije se prave pomoću komande *FLUSH TABLES WITH READ LOCK* da bi se napravio snimak baze

podataka (eng. *snapshot*). Time se sprečava upis u bazu, obično na nekoliko sekundi. Ako postoji operacija na čekanju u vreme pokušaja izrade rezervne kopije, Google Cloud SQL pokušava ponovo dok se prozor rezervne kopije ne završi [6]. Operacije koje blokiraju sigurnosne kopije su dugotrajne operacije kao što su uvoz, izvoz, ažuriranje i ponovno pokretanje instance.

Za razliku od automatskih kopija koje se kreiraju u zadato vreme, kopije na zahtev mogu da se naprave u bilo kom trenutku. To je korisno kod izvršavanja rizičnih operacija u bazi podataka ili ako je potrebno da se kreira rezervna kopija pre prozora za pravljenje automatskih kopija. Rezervne kopije na zahtev se ne brišu periodično, već su trajne, sve dok se ne izbrišu eksplicitno ili dok se ne izbriše sama instanca.

Za upravljanje rezervnim kopijama koriste se *gcloud* komande u obliku *gcloud sql backups COMMAND [GLOUD_WIDE_FLAG ...]*. One omogućavaju izlistavanje i pribavljanje informacija o rezervnim kopijama za Cloud SQL instancu. Komanda može da bude *create*, *delete*, *restore*, *list* ili *describe* [4].

2.1.2. Replikacija

Cloud SQL ima mogućnost replikacije ako se koristi MySQL ili PostgreSQL baza podataka. Replikacija je mehanizam kopiranja podataka u više instanci čime se postiže redundantnost podataka. Time se ublažavaju kvarovi sistema. U stvari, replikacija omogućava visoko dostupan sistem. CloudSQL ima klaster okruženje za replikaciju koje se sastoji od primarne (ili glavne) instance, instance u stanju pripravnosti i jedne ili više replika za čitanje. Primarni razlog za korišćenje replikacije je skaliranje korišćenja podataka u bazi bez pogoršanja performansi. Drugi razlozi uključuju migraciju podataka između regiona ili platformi, kao i migraciju iz neke privatne baze podataka na Cloud SQL [6].

Kada se odnosi na Cloud SQL instancu, replicirana instanca naziva se primarna instanca, a kopije replike za čitanje. Sve primarne instance i replike za čitanje nalaze se u Cloud SQL-u. Kada se odnosi na lokalnu bazu podataka, scenario replikacije naziva se replikacija sa spoljnog servera. U ovom scenariju baza podataka koja se replicira je izvorni server. Kopije koje se nalaze u Cloud SQL-u nazivaju se Cloud SQL replike. Takođe, postoji instanca koja predstavlja izvorni server baze podataka u Cloud SQL-u, a zove se instanca reprezentacije izvora (eng. *source representation instance*).

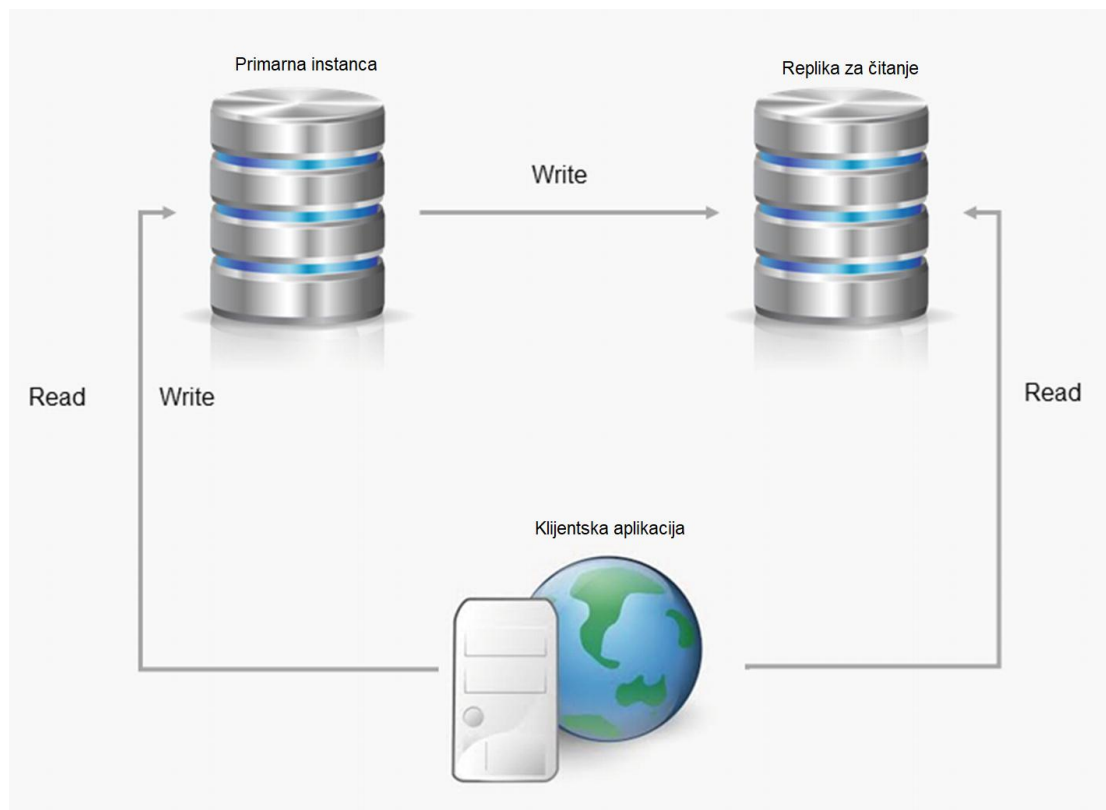
Cloud SQL podržava sledeće tipove replika [6]:

- replike za čitanje
- međuregionalne replike za čitanje
- eksterne replike za čitanje kod MySQL-a
- Cloud SQL replike, kada se repliciraju sa spoljnog servera, takođe kod MySQL-a.

Replike za čitanje se koriste kako bi se rasteretila instanca Cloud SQL-a. Ove replike su tačne kopije primarne instance. Podaci i druge promene na primarnoj instanci ažuriraju se u gotovo realnom vremenu na replikama za čitanje. Kao što njihov naziv nagoveštava, replike su samo za čitanje, tj. nemoguć je upis. Replika za čitanje obrađuje upite i zahteve za čitanje smanjujući tako opterećenje na primarnoj instanci. Maksimalan broj replika za čitanje po primarnoj instanci je 10. Preporučljivo je postaviti replike za čitanje u zonu različitu od zone primarne instance jer se time osigurava da replike za čitanje neće prekinuti sa radom kada zona koja sadrži primarnu instancu padne. Slika 4 ilustruje postojanje replike za čitanje uz primarnu instancu.

Da bi kreiranje replike za čitanje primarne instance Cloud SQL-a bilo moguće, ona mora da ispunjava sledeće zahteve:

- automatske rezervne kopije moraju da budu omogućene,
- binarna evidencija mora biti omogućena, što zahteva da bude omogućen i *point-in-time* oporovak (vraćanje na neko prethodno stanje),
- nakon omogućavanja binarnog evidentiranja mora da bude kreirana bar jedna rezervna kopija.



Slika 3: Replike za čitanje

Replikacija između različitih regiona omogućava pravljenje replike za čitanje u drugoj regiji u odnosu na primarnu instancu [6]. Međuregionalne replike za čitanje imaju sledeće benefite:

- poboljšavaju performanse čitanja, jer korisnik može da postavi repliku bliže regionu u kojem je aplikacija,
- obezbeđuju dodatnu sposobnost oporavka od katastrofe ukoliko dođe do kvara na nivou regiona,
- omogućavaju migraciju podataka iz jedne regije u drugu sa minimalnim zastoјima.

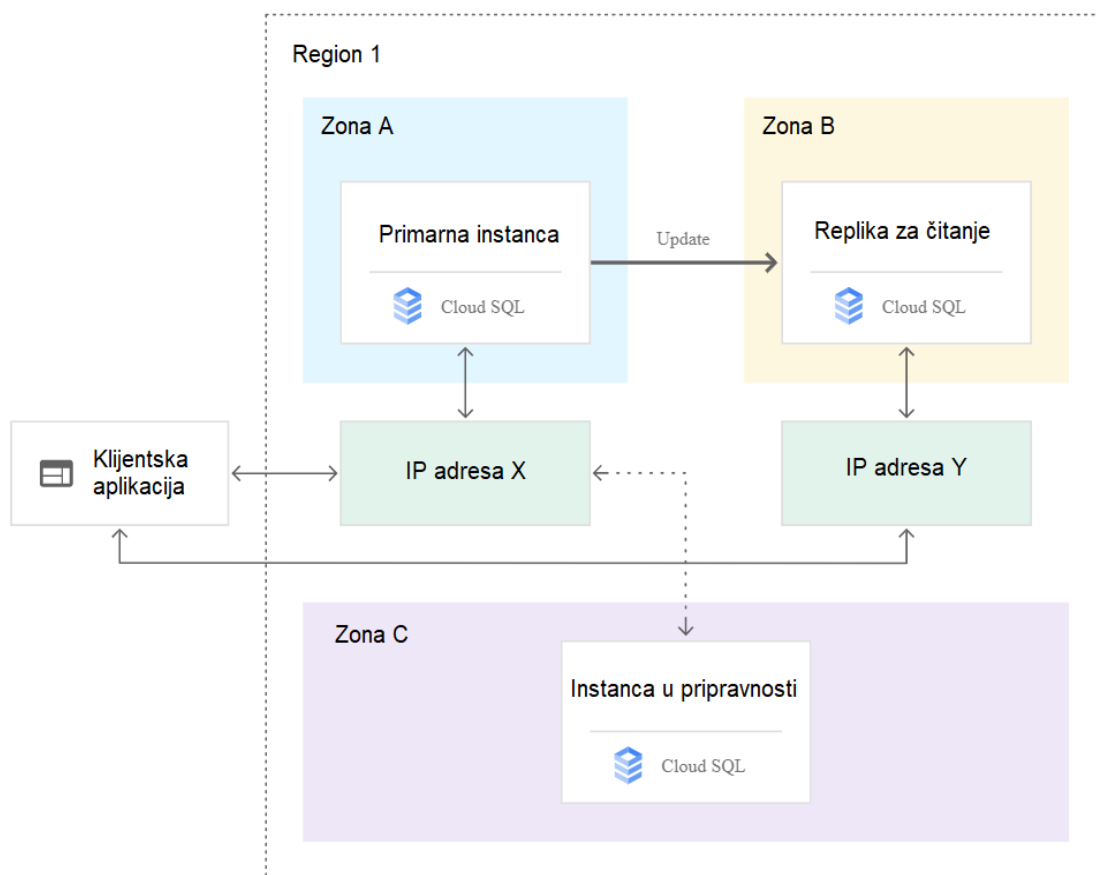
Eksterne replike za čitanje se koriste kod MySQL baze podataka i to su eksterne instance koje se repliciraju iz primarne instance Cloud SQL-a (npr. MySQL instanca koja se izvršava na Compute Engine-u).

Ako se kao baza koristi PostgreSQL, Cloud SQL omogućava i logičku replikaciju. Logička replikacija je fleksibilna i osim što omogućava standardnu replikaciju podataka sa primarne instance, moguća je i selektivna replikacija, tj. replikacija samo određenih tabela ili čak vrsta iz tabele [6]. Pored toga, logičkom replikacijom podaci mogu da se iskopiraju i na bazu podataka koja nije PostgreSQL.

2.1.3. Visoka dostupnost i *failover*

Prilikom kreiranja instanci, moguće je izabrati lokaciju na kojoj će ona da se nalazi. Lokacija može da bude regionalna, tj. da se odnosi na jednu geografsku lokaciju, ili multiregionalna. Zone su podlokacije u okviru nekog regiona. Svrha konfiguracije visoke dostupnosti (eng. *high availability*, HA) je da smanji zastoје kada zona ili instanca postanu nedostupne. To se može dogoditi tokom prekida rada zone, ili kad dođe do oštećenja instance. Uz HA, podaci su i dalje dostupni klijentskim aplikacijama. Konfiguracija HA, koja se ponekad naziva i klaster, omogućava redundantnost podataka. Instanca Cloud SQL-a konfigurisana za HA takođe se naziva regionalna instanca i nalazi se u primarnoj i sekundarnoj zoni unutar konfigurisane regije [6]. Unutar regionalne instance, konfiguraciju čine primarna instanca i instanca u stanju pripravnosti (eng. *stand by instance*) koja se kreira automatski, transparentno za korisnika, kada je HA omogućena. Kroz sinhronu replikaciju na trajni disk svake zone, svi upisi napravljeni u primarnu instancu repliciraju se na diskove u obe zone pre nego što se transakcija prijavi kao izvršena. U slučaju otkazivanja instance ili zone, trajni disk se povezuje na instancu u pripravnosti i postaje nova primarna instance, na koju se preusmeravaju korisnici. Ovaj proces se naziva preusmeravanje (eng. *failover*). Nakon *failover*-a, instanca koja je preuzela ulogu primarne nastavlja da bude primarna instanca, čak i nakon što se originalna instanca vrati na mrežu. Kada zona ili instanca koja je pala ponovo postane dostupna, prvobitna primarna instanca se uništava i ponovo kreira i postaje nova instanca u stanju pripravnosti. Ako instanca konfigurisana za HA prestane da reaguje, Cloud SQL se automatski prebacuje na dostavljanje podataka iz instance u stanju pripravnosti [6].

Replika za čitanje je kopija primarne instance. Podaci i druge promene na primarnoj instanci ažuriraju se u gotovo realnom vremenu na replici. Replike za čitanje ne mogu da budu dostupne kao primarne instance. Tokom prekida zone, zaustavlja se saobraćaj do replika za čitanje u toj zoni. Kada zona ponovo postane dostupna, sve replike za čitanje u zoni nastavljaju replikaciju iz primarne instance. Ako su u zoni koja nije u prekidu, povezane su sa instancom u stanju pripravnosti kada postane primarna instanca. Dobra praksa je da replike za čitanje budu u različitoj zoni u odnosu na primarnu instancu i instancu u pripravnosti. Na primer, ako je primarna instanca u zoni A, a *stand-by* u zoni B, replike za čitanje treba staviti u zonu C. Ovim se osigurava da će replike nastaviti da rade čak i ako zona u kojoj je primarna instanca padne. Na slici 4 dat je prikaz konfiguracije Cloud SQL instance sa replikom za čitanje u različitoj zoni, u normalnom režimu rada.



Slika 4: HA konfiguracija u normalnom režimu

Da bi Cloud SQL omogućio preusmeravanje, konfiguracija mora da bude sledeća [6]:

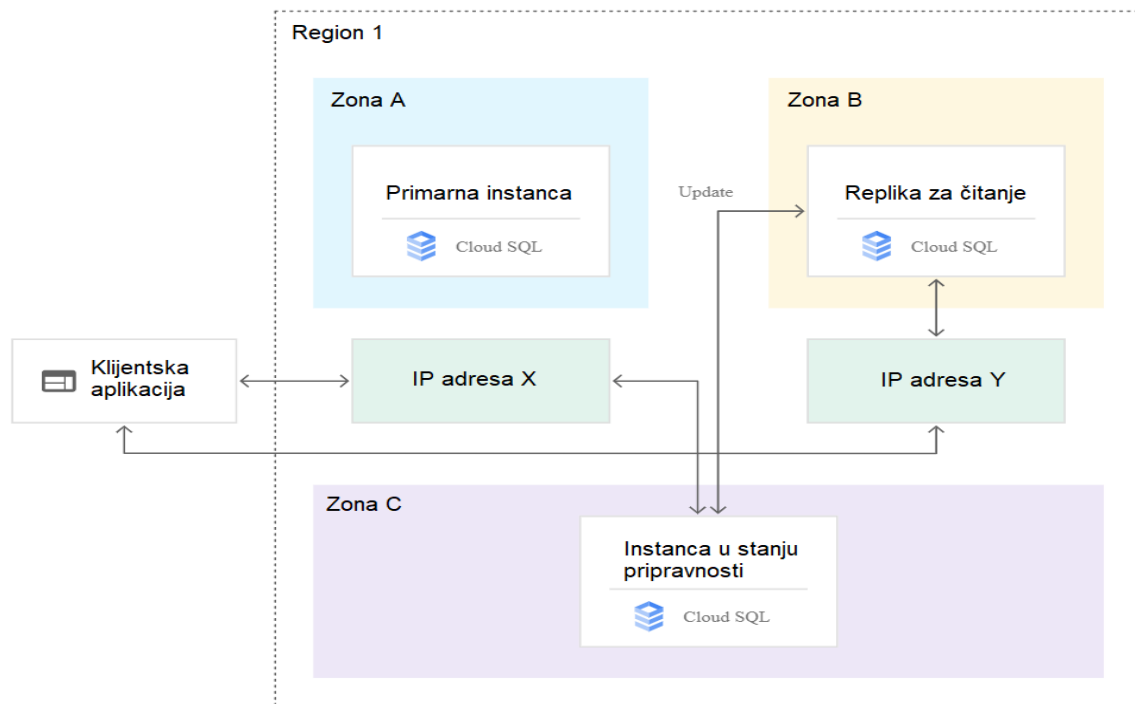
- Primarna instanca mora biti u normalnom operativnom stanju (nije zaustavljena ili na održavanju i ne izvršava dugotrajnu operaciju poput pravljenja sigurnosne kopije, uvoza ili izvoza).

- Sekundarna zona i instanca u stanju pripravnosti moraju biti u zdravom stanju. Kada instanca u stanju pripravnosti ne reaguje i/ili je replikacija na sekundarnu zonu prekinuta, operacije preusmeravanja se blokiraju. Nakon što Cloud SQL popravi stanje pripravnosti i sekundarna zona postane dostupna, replikacija se nastavlja i moguć je *failover*.

U slučaju pada primarne instance ili zone, odvija se sledeći proces:

1. Primarna instanca ili zona ne radi ispravno. Svake sekunde primarna instanca vrši upis u sistemsku bazu podataka, kao signal otkucaja srca. Ako se ne detektuje više takvih otkucaja, pokreće se preusmeravanje. To se dešava ako primarna instanca ne reaguje približno 60 sekundi ili ako zona koja sadrži primarnu instancu prekine sa radom
2. Nakon ponovnog povezivanja sistema, instanca u stanju pripravnosti služi za dostavljanje podataka. Kroz deljenu statičku IP adresu sa primarnom instancom, instanca u stanju pripravnosti sada dostavlja podatke iz sekundarne zone.

Na slici 5 dat je prikaz instance kad se izvršava preusmeravanje. Nakon njega, primarna instanca i instanca u stanju pripravnosti menjaju uloge, što znači da sada instanca u stanju pripravnosti vrši ažuriranje replike za čitanje.



Slika 5: Preusmeravanje (*failover*)

Ako je potrebno da primarna instanca radi u zoni u kojoj je došlo do prekida, treba izvršiti vraćanje (eng. *failback*). Izvršava se identično kao preusmeravanje, ali u suprotnom smeru, da bi se saobraćaj preusmerio na originalnu instancu.

2.1.4. Oporavak od katastrofe

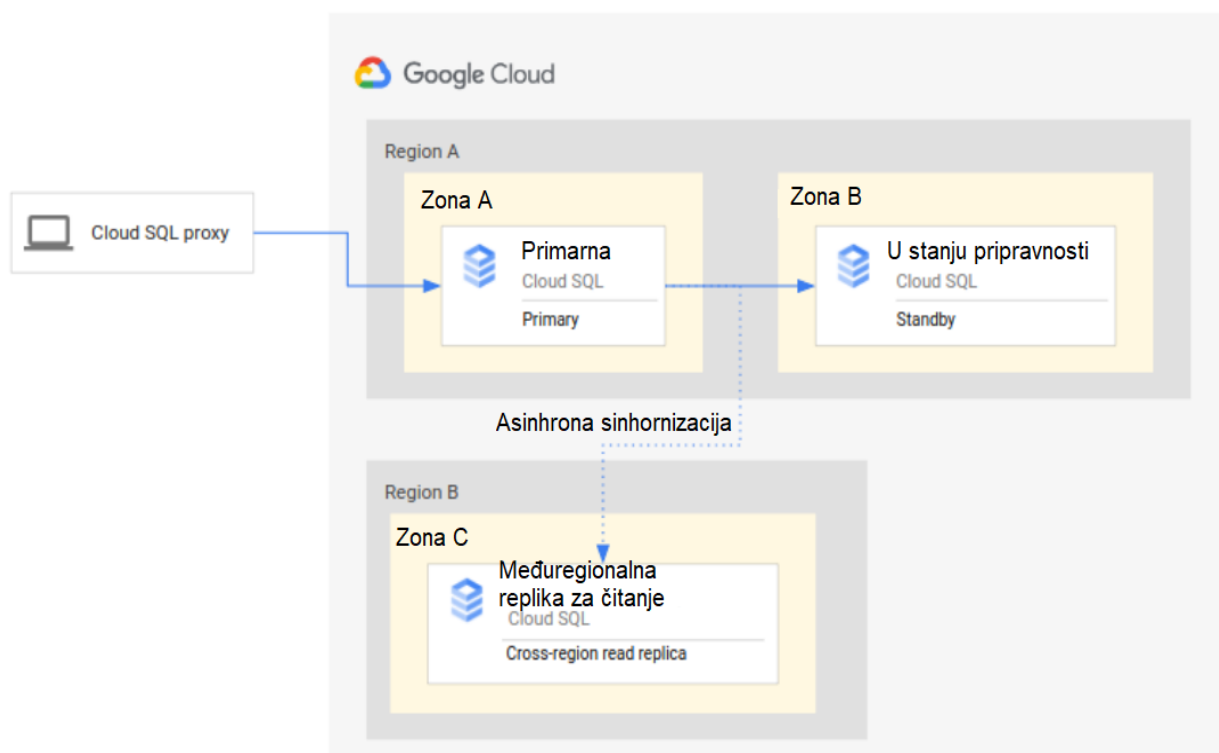
U Google Cloud-u, oporavak od katastrofe baze podataka (eng. *disaster recovery*, *DR*) podrazumeva obezbeđivanje kontinuiteta obrade, posebno kada region zakaže ili postane nedostupan [7]. Ako Google Cloud region koja hostuje Cloud SQL bazu podataka postane nedostupan, tada i SQL baza podataka postaje nedostupna. Da bi se obrada podataka nastavila neometeno, baza podataka treba da bude dostupna u sekundarnom regionu što je pre moguće. Plan DR za Cloud SQL zahteva da se konfigurišu replike za čitanje u više regiona u Cloud SQL-u. Da bi DR bio moguć, Cloud SQL instanca mora da ima omogućenu visoku dostupnost.

Generalno postoje dve varijante procesa DR:

- Primarni region pada i baza podataka prelazi u sekundarni region. Kad baza postane spremna i neka aplikacija počne da je koristi, ona postaje nova primarna baza podataka i ostaje primarna do daljnjeg.
- Nakon pada baza prelazi u sekundarni region, ali kad primarni region ponovo postane dostupan, baza se vraća u njega. Ova varijanta DR procesa je posebno relevantna za baze podataka koje moraju da se izvršavaju u primarnom regionu zbog kašnjenja mreže ili zato što su neki resursi dostupni samo u primarnom regionu. U ovom slučaju baza podataka radi u sekundarnom regionu samo za vreme prekida u primarnom regionu.

Na slici 5 prikazana je arhitektura koja podržava oporavak od katastrofe kod Cloud SQL instance. Arhitektura ima dve instance (primarna instanca i instanca u stanju pripravnosti) koje se nalaze u dve odvojene zone u okviru jedne regije (primarna regija). Sinhronizacija instanci vrši se preko regionalnih trajnih diskova. Pored toga, jedna instanca Cloud SQL-a (međuregionalna replika za čitanje) nalazi se u drugoj regiji (sekundarnoj regiji). Za DR, replika za čitanje između različitih regiona podešena je za sinhronizaciju (korišćenjem asinhronne replikacije) sa primarnom instancom. Primarna i instanca u stanju pripravnosti dele isti regionalni disk, pa su njihova stanja identična. Pošto se koristi asinhrona replikacija, moguće je da međuregionalna replika za čitanje zaostaje za primarnom instancom. Kada se desi preusmeravanje, međuregionalna replika za čitanje može da podrži RPO¹ (eng. *Recovery Point Objective*) od 0.

¹ RPO je maksimalna količina podataka koja može da se izgubi, a da ne dođe do oštećenja sistema.



Slika 6: DR arhitektura

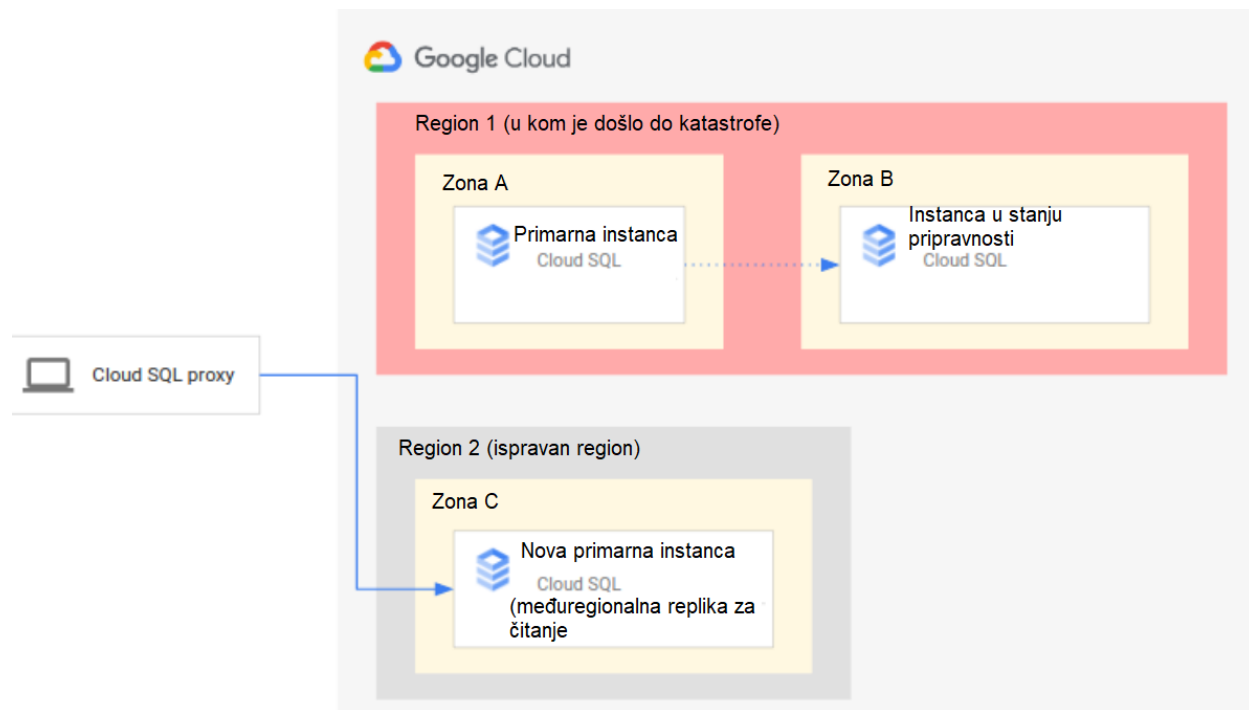
Osnovni DR proces započinje kada primarni region postane nedostupan, a primarna instanca ne uspe da nastavi obradu u sekundarnoj regiji. DR propisuje operativne korake koji moraju da se izvrše, ručno ili automatski, kako bi se ublažio regionalni kvar i da bi se uspostavila primarna instanca koja će da se izvršava u sekundarnoj regiji. Na slici 7 dat je prikaz DR procesa koji se sastoji od sledećih koraka [7]:

1. Primarni region (R1), u kom se izvršava primarna instanca, postaje nedostupan.
2. Operativni tim prepoznaje i beleži da je došlo do katastrofe i odlučuje da li je potrebno preusmeravanje.
3. Ako je preusmeravanje potrebno, međuregionalna replika za čitanje u sekundarnom regionu (R2) postaje nova primarna instanca.
4. Klijentske veze su ponovo konfigurisane da nastave obradu na novoj primarnoj instanci i pristupaju primarnoj instanci u sekundarnom regionu.

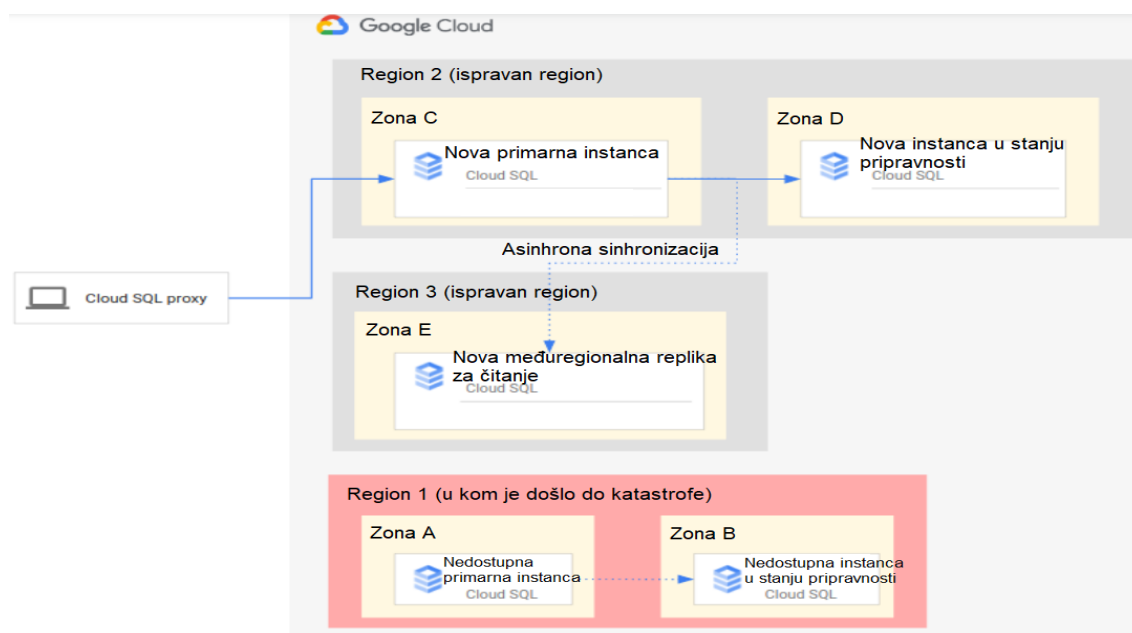
Ovim se postiže ispravan rad primarne instance baze podataka, ali se ne uspostavlja potpuna DR arhitektura, u kojoj primarna instanca ima instancu u stanju pripravnosti i repliku za čitanje.

Za razliku od osnovnog, kompletan DR proces garantuje da je nova primarna instanca, omogućena za HA i ima međuregionalna repliku za čitanje. On je proširenje osnovnog procesa i ima dodatne korake. Nakon izvršenja svih koraka iz osnovnog procesa, kreira se nova instanca u stanju pripravnosti u sekundarnom regionu, ali u različitoj zoni od one u kojoj je primarna, povezuje se sa primarnom instancom i kreće sa izvršenjem. U trećem regionu kreira se međuregionalna replika

za čitanje i vezuje se na primarnu instancu čime je uspostavljena kompletna operativna DR arhitektura. Na slici 8 prikazana je kompletna DR arhitektura nakon preusmeravanja.



Slika 7: Osnovni DR process



Slika 8: Kompletan DR proces

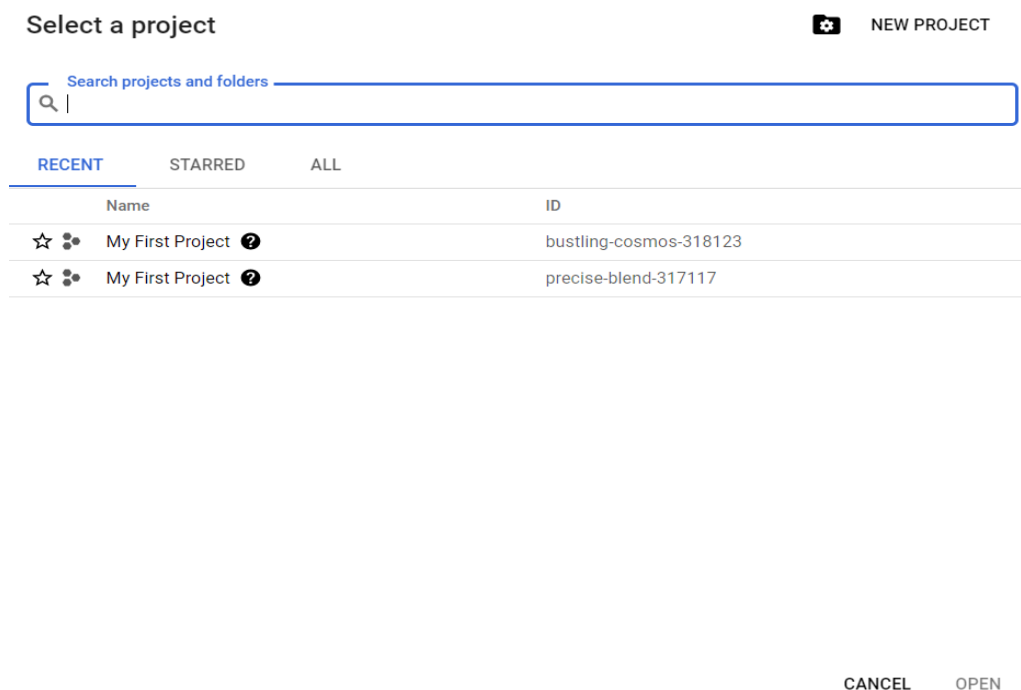
Kad primarni region prestane sa radom, to ne znači da se primarna instanca i instanca u stanju pripravnosti koje mu pripadaju automatski isključuju ili postaju nedostupne kad region ponovo postane dostupan. Kao posledica toga, korisnici mogu da čitaju ili upisuju podatke u staru primarnu instancu, umesto u novu koja je nastala DR procesom. Zbog toga je dobra praksa onemogućiti pristup originalnoj primarnoj instanci kad se kreira nova, a onda je i obrisati [7].

3. MySQL na Cloud SQL-u

MySQL je jedan od tri relacione baze podataka kojima je moguće upravljati pomoću Google Cloud SQL-a. Podrazumevana verzija MySQL-a na Cloud SQL-u je 5.7, a podržane su i 5.6 i 8.0. Dostupno je to 624GB RAM memorije i 30TB skladišta uz mogućnost automatskog povećanja veličine skladišta kada je to potrebno. Kreiranje i upravljanje MySQL instancom baze podataka moguće je iz Google Cloud konzole, a pristup bazi može biti kako sa privatnih adresa, tako i iz eksternih aplikacija. Moguće je kreiranje automatskih rezervnih kopija, replikacija, podešavanje sigurnog povezivanja na instancu i sl. Primeri kreiranja instance i podešavanja biće opisani u nastavku.

3.1. Kreiranje instance

Kreiranje instance može da se obavi preko konzole ili *gcloud*-a. Najpre je potrebno izabrati projekat u okviru kog će da se kreira MySQL instanca (slika 9):



Slika 9: Odabir projekta

Po zatvaranju prethodnog dijaloga, u gornjem delu stranice vidi se ime izabranog projekta, a na stranici su prikazane informacije o projektu. Drugi korak neophodan za kreiranje instance je omogućavanje *Compute Engine API*-ja. Compute Engine API pruža korisnicima interfejs za upravljanje resursima (projekti, instance i sl). Nakon toga, moguće je kreirati instancu i pojavljuje

se dijalog kao na slici 10. Prilikom kreiranje instance, potrebno je obeležiti koje funkcionalnosti se zahtevaju od instance. Obavezno je uneti jedinstveni identifikator instance, lozinku za pristup, region u kom se nalazi (podešava se samo jednom, prilikom kreiranja instance) i verziju MySQL baze. Verzija baze podataka ne može da se promeni nakon kreiranja instance.

Instance info

Instance ID *
mysql1135

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password *
••••••••

Set a password for the root user. [Learn more](#)

☐ No password

Database version *
MySQL 8.0

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region
us-central1 (Iowa)

☐ Single zone
In case of outage, no failover. Not recommended for production.

☒ Multiple zones (Highly available)
Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost. Enables binary logs (required for replication) and automatic backups. Make sure your storage can support at least 7 days of logs.

Slika 10: Kreiranje instance

Pored ovih stvari, moguće je konfigurisati dodatne funkcionalnosti poput veličine skladišta, održavanja, automatskog kreiranja rezervnih kopija i automatskog skaliranja. Odabirom konkretne stavke, podešavaju se parametri vezani za tu funkcionalnost (npr. podešavanje četvoročasovnih prozora za kreiranje automatskih rezervnih kopija).

Drugi način za kreiranje instance je iz *gcloud*-a. Na slici 11 dat je primer kreiranja instance Cloud SQL-a iz *gcloud*-a. Dodatni parametri podešavaju se preko flegova, a oni koji se ne navedu postavljaju se na podrazumevane vrednosti.

```
cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances create mysqlshell1135 --cpu=1 --memory=4 --region=us-central1
Creating Cloud SQL instance...!
```

Slika 11: Kreiranje instance iz *gcloud*-a

3.2. Povezivanje na instancu

Pomoću komande *gcloud sql instances list* izlistavaju se sve kreirane Cloud SQL instance u projektu (slika 12). Sve *gcloud* naredbe imaju prefiks *gcloud sql*.

```
@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances list
```

NAME	DATABASE_VERSION	LOCATION	TIER	PRIMARY_ADDRESS	PRIVATE_ADDRESS	STATUS
mysqlshell11135	MYSQL_5_7	us-central1-f	db-custom-1-4096	34.133.144.127	-	RUNNABLE
mysql11135	MYSQL_8_0	us-central1-f	db-custom-4-26624	34.133.255.102	-	RUNNABLE

```
eonrokk123@cloudshell:~ (bustling-cosmos-318123)$
```

Slika 12: Izlistavanje Cloud SQL instanci

Pošto su obe instance pokrenute (u stanju *RUNNABLE*), moguće je povezati se na neku od njih. Povezivanje se vrši pomoću komande *connect*. Na slici 13 prikazano je povezivanje iz komandne linije na prvu kreiranu instancu. Povezivanje je potrebno da bi se kreirala baza podataka.

```
cloudshell:~ (bustling-cosmos-318123)$ gcloud sql connect mysql11135 --user=root --quiet
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 177
Server version: 8.0.18-google (Google)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Slika 13: Povezivanje na instancu

Osim povezivanja iz Google komandne linije, instanca može da se konfigurise za pristup od spolja. Tačnije, moguće je podešavanje pristupa sa privatnih adresa koje pripadaju virtuelnom privatnom oblaku i javnih adresa, preko Cloud SQL Proxy-ja (svi podaci su šifrovani), drugih Google Cloud servisa, kao i iz MySQL Workbench-a².

Sada kada je izvršeno povezivanje, baza podataka i tabele prave se istim komandama kao u MySQL-u. Ukoliko korisnik ne poznaje MySQL, bazu može da kreira i samo preko *gcloud* komandi, što je ilustrovano na slici 14.

² Nije bilo moguće testirati jer IP adresa ne pripada VPC-u, a ne spada u javne.

```

eonrokk123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql databases create test --instance=mysqlshell1135 -i mysqlshell1135
Creating Cloud SQL database...done.
Created database [test].
instance: mysqlshell1135
name: test
project: bustling-cosmos-318123

```

Slika 14: Kreiranje baze iz gcloud-a

Povezivanjem na MySQL instancu korisnik dobija MySQL prompt i može da napravi bazu podataka, tabele i ubaci podatke. Na slici 15 prikazano je kreiranje *guestbook* baze podataka.

```

mysql> CREATE DATABASE guestbook;
Query OK, 1 row affected (0.13 sec)

mysql>
mysql> USE guestbook;
Database changed
mysql>
mysql> CREATE TABLE entries (guestName VARCHAR(255), content VARCHAR(255), entryID INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(entryID));
Query OK, 0 rows affected (0.14 sec)

mysql>
mysql> INSERT INTO entries (guestName, content) values ("first guest", "I got here!");
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO entries (guestName, content) values ("second guest", "Me too!");
Query OK, 1 row affected (0.12 sec)

mysql> commit;
Query OK, 0 rows affected (0.10 sec)

mysql>

```

Slika 15: Kreiranje tabele

U suštini, MySQL promptu mogu da se zadaju uglavnom sve MySQL komande koje bi se inače koristile u MySQL Workbench-u ili klijentu. Ipak, neke od njih nisu još uvek integrisane, pa ne mogu da se koriste:

- *LOAD DATA INFILE*
- *SELECT ... INTO OUTFILE*
- *SELECT ... DUMPFILE*
- *INSTALL/UNINSTALL PLUGIN*
- *CREATE FUNCTION ... SONAME ...*
- *LOAD_FILE()* funkcija.

Što se tiče MySQL-a 8.0, zbog promena u njemu, ne mogu da se koriste naredbe *CREATE* ili *DELETE* za upravljanje korisnicima. Takođe, korisničke privilegije nije moguće promeniti naredbama *INSERT*, *UPDATE* ili *DELETE*. Umesto toga koriste se naredbe *CREATE/DROP USER* i *GRANT /REVOKE* za dodelu ili ukidanje privilegija. Za MySQL 8.0. dozvoljena je i naredba *CREATE TEMPORARY TABLE* (za verziju 5.6. i 5.7. nije).

3.3. Kreiranje rezervnih kopija

Kao što je već pomenuto, postoje 2 tipa rezervnih kopija, automatske i kopije koje se kreiraju na zahtev korisnika. Kreiranje kopija na zahtev može da se izvrši iz komandne linije ili otvaranjem kartice *Backups*, što je prikazano na slici 14³:

Create a Backup

Backups allow you to restore instances to recover lost data or undo errors.

To help you save money, backups work incrementally. Each backup stores only the changes to your data since the previous backup. [Learn more](#)

i You'll be billed \$0.08/GB per month for data storage.

Describe this backup (optional)

You can make a note here to help you identify this backup. 0 / 140

Choose where to store your backups

Backups are stored in the closest multi-region location to you by default. Only customize if needed.

☒ Multi-region (default)

☐ Region

Location *

us - Data centers in the Un...

LOCATION OPTIONS

CREATE CANCEL

```
cloudshell:~ (bustling-cosmos-318123)$ gcloud sql backups create --async --instance [mysql1135]
```

Slika 16: Kreiranje on-demand kopije

Kad su u pitanju automatske rezervne kopije, ako su dozvoljene, moguće je podesiti vremenski period (prozor) u kom će da se kreiraju. To je takođe dostupno preko kartice *Backups* ili iz komandne linije (slika 15):

```
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch mysql1135 --backup-start-time 03:00
```

The following message will be used for the patch API method.

```
{
  "name": "mysql1135",
  "project": "bustling-cosmos-318123",
  "settings": {
    "backupConfiguration": {
      "backupRetentionSettings": {
        "retainedBackups": 7,
        "retentionUnit": "COUNT",
        "binaryLogEnabled": true,
        "enabled": true,
        "location": "us",
        "startTime": "03:00",
        "transactionLogRetentionDays": 7
      }
    }
  }
}
```

Patching Cloud SQL instance...done.

Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/mysql1135].

```
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$
```

Slika 17: Podešavanje automatskih rezervnih kopija

Sve napravljene rezervne kopije izlistavaju se komandom *gcloud sql backups list*, a moguće je pregledati informacije o konkretnoj kopiji korišćenjem *describe*. Na slici 16 prikazan je primer korišćenja ovih komandi.

³ Ova opcija se naplaćuje dodatno, pa nije isprobana, već je samo prikazan interfejs za kreiranje kopija na zahtev.

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql backups list --instance mysql1135
ID          WINDOW_START_TIME  ERROR  STATUS  INSTANCE
1624752832852  2021-06-27T00:13:52.852+00:00  -      SUCCESSFUL  mysql1135
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql backups describe 1624752832852 --instance mysql1135
backupKind: SNAPSHOT
description: Taking a backup after instance creation
endTime: '2021-06-27T00:14:33.489Z'
enqueueTime: '2021-06-27T00:13:52.852Z'
id: '1624752832852'
instance: mysql1135
kind: sql#backupRun
location: us-centrall
selfLink: https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/mysql1135/backups/1624752832852
startTime: '2021-06-27T00:13:52.853Z'
status: SUCCESSFUL
type: ON_DEMAND
windowStartTime: '2021-06-27T00:13:52.852Z'

```

Slika 18: Pregled informacija o napravljenim kopijama

3.4. Kreiranje replika za čitanje

Da bi se kreirala replika za čitanje, instanca čija se replikacija vrši mora da ispuni neke uslove, tj. da bude primarna i da je omogućena replikacija i binarno logovanje. Ako je to ispunjeno, što se vidi pozivom komande *describe* nad instancom, može da se kreira replika. Na slici 17 prikazane su informacije o instanci *mysql1135*. Polje *type* označava da je u pitanju primarna instanca, a ne replika, što je važno jer nije moguće kreirati replike nad replikama. Iz označenog *backupConfiguration* polja vidi se da je binarno logovanje omogućeno, kao i da je kreiranje automatskih kopija podešeno. Sada se replika za čitanje kreira odgovarajućom *gcloud* komandom, prikazanom na slici 18. Kreiranje se odvija kao i kreiranje obične instance, ali se navodi primarna instanca na koju se replika odnosi.

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances describe mysql1135
backendType: SECOND_GEN
connectionName: bustling-cosmos-318123:us-centrall:mysql1135
databaseVersion: MYSQL_8_0
etag: c769840669719d8d607c3fb14622971a2e26a225abeecd8bd718923e30288474
failoverReplica:
  available: true
gceZone: us-centrall-f
instanceType: CLOUD_SQL_INSTANCE
ipAddresses:
- ipAddress: 34.133.255.102
  type: PRIMARY
  ...
  ...
  ...
backupConfiguration:
  backupRetentionSettings:
    retainedBackups: 7
    retentionUnit: COUNT
  binaryLogEnabled: true
  enabled: true

```

Slika 19: Uslovi za kreiranje replike za čitanje

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances create testreplica --master-instance-name=mysql1135
WARNING: Starting with release 233.0.0, you will need to specify either a region or a zone to create an instance.
Creating Cloud SQL instance...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/testreplica].
NAME          DATABASE_VERSION LOCATION    TIER      PRIMARY_ADDRESS PRIVATE_ADDRESS STATUS
testreplica    MYSQL_8_0        us-central1-f db-custom-4-26624 34.132.55.238 -          RUNNABLE

```

Slika 20: Kreiranje replike za čitanje

Informacije o replikama dobijaju se na isti način kao i za primarne instance, komandom `gcloud sql instances describe imeReplike`.

3.5. HA konfiguracija

Prilikom kreiranja instance, postoji opcija za omogućavanje HA konfiguracije. To može da se izvrši i naknadno iz konzole, komandom prikazanom na slici 19:

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch mysql1135 --availability-type REGIONAL --enable-bin-log --backup-start-time 02:00
The following message will be used for the patch API method.
{"name": "mysql1135", "project": "bustling-cosmos-318123", "settings": {"availabilityType": "REGIONAL", "backupConfiguration": {"backupRetentionSettings": {"retainedBackups": 7, "retentionUnit": "COUNT"}, "binaryLogEnabled": true, "enabled": true, "location": "us", "startTime": "02:00", "transactionLogRetentionDays": 7}}}
Patching Cloud SQL instance...done.
Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/mysql1135].

```

Slika 21: Omogućavanje HA konfiguracije

Omogućavanjem visoke dostupnosti, CloudSQL automatski kreira dodatnu instancu u stanju pripravnosti. Iz komandne linije, nakon podešava HA konfiguracije, može da se inicira preusmeravanje (*failover*) da bi se proverilo kako će aplikacija da reaguje ako dođe do toga. Dok se komanda izvršava, master instanca nije dostupna i nije moguće povezati se na nju dok se preusmeravanje ne izvrši. Ako je preusmeravanje uspešno, nakon njegovog inciranja korisnik može uspešno da se poveže na instancu, tj. izvršen je i *failback*, što je ilustrovano na slici 20.

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances failover mysql1135
Failover will be initiated. Existing connections to the master
instance will break and no new connection can be established during
the failover.

Do you want to continue (Y/n)? Y

Failing over Cloud SQL instance...done.
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql connect mysql1135 --user=root --quiet
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 95
Server version: 8.0.18-google (Google)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Slika 22: Iniciranje preusmeravanja

3.6. Simulacija DR procesa

Oporavak od katastrofe može da se izvrši pomoću međuregionalnih replika. Najpre je potrebno podesiti visoku dostupnost, što je prikazano u prethodnom potpoglavlju. Kreira se nova replika za čitanje, a onda se vrši povezivanje na nju i proverava da li replikacija radi, tj. da li je *guestbook* baza kreirana u primarnoj instanci vidljiva iz replike (slika 23):

```
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql connect testreplica --user=root --quiet
Allowlisting your IP for incoming connection for 5 minutes...done.

Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1915
Server version: 8.0.18-google (Google)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

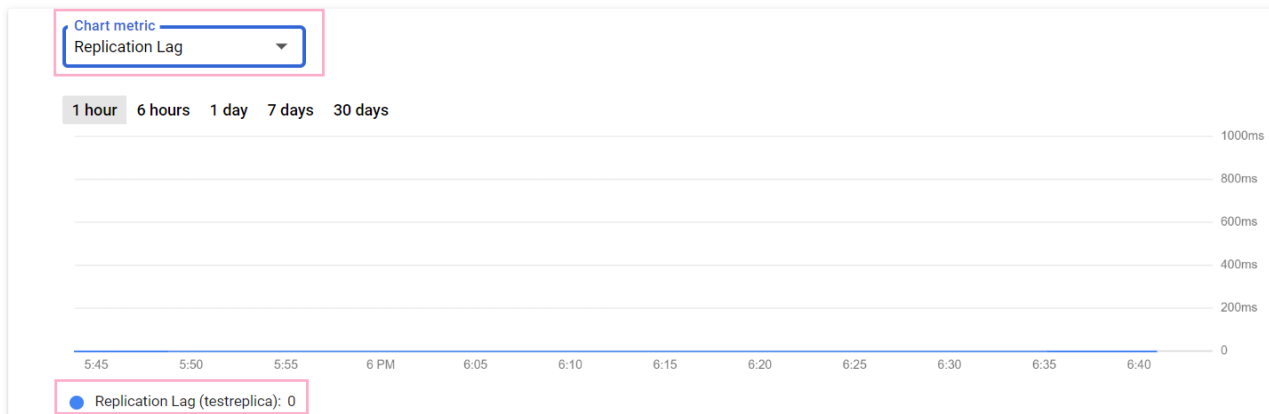
mysql> use guestbookč
ERROR 1049 (42000): Unknown database 'guestbookč'
mysql> use guestbook;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_guestbook |
+-----+
| entries              |
+-----+
1 row in set (0.11 sec)
```

Slika 23: Provera replikacije

Pošto je utvrđeno da replikacija radi ispravno, sledi faza u kojoj se simulira prekid rada primarnog regiona (nedostupnost primarne baze podataka). Proverava se zaostajanje replike i u idealnom slučaju, kad se desi prekid primarnog regiona zaostojanje je nula, tj. sve transakcije se repliciraju. Ako nije nula, neke transakcije se možda neće replicirati i tada međuregionalna replika za čitanje ne sadrži sve transakcije izvršene na primarnoj instanci. Zaostajanje se vidi odabirom *Replication Lag* stavke iz menija *Chart Metric* i sa slike 24 se vidi da replika *testreplica* ne zaostaje za primarnom instancom.

Da bi se simulirala katastrofa, primarna instanca mora da bude nedostupna, odnosno, treba da se stopira. Zato je prvo potrebno raskinuti vezu (eng. *detach*) između replike za čitanje i primarne instance, jer inače zaustavljanje primarne instance neće biti moguće. Komanda za raskidanje veze i zaustavljanje primarne instance prikazana je na slici 25.



Slika 24: Replication lag

```
eonrokkrl123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch testreplica --no-enable-database-replic
The following message will be used for the patch API method.
{"name": "testreplica", "project": "bustling-cosmos-318123", "settings": {"databaseReplicationEnabled": false}}
WARNING: This patch modifies a value that requires your instance to be
restarted. Submitting this patch will immediately restart your
instance if it's running.

Do you want to continue (Y/n)? Y

Patching Cloud SQL instance...done.
Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/testreplica].
eonrokkrl123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch mysql1135 --activation-policy NEVER
The following message will be used for the patch API method.
{"name": "mysql1135", "project": "bustling-cosmos-318123", "settings": {"activationPolicy": "NEVER"}}
Patching Cloud SQL instance...done.
Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/mysql1135].
```

Slika 25: Zaustavljanje primarne instance

Sada se startuje DR proces. Međuregionalna replika za čitanje se unapređuje u samostalnu (eng. *standalone*) instancu i sada je ona nova primarna instanca, dok je originalna primarna zaustavljena. Pokreće se nova primarna instanca i omogućavanjem HA konfiguracije i kreiranjem replike za čitanje dobija se početna arhitektura koja ispravno radi. Nakon toga, treba obrisati staru primarnu instancu da ne bi došlo do toga da neki korisnik slučajno upisuje ili čita s nje. Ovi koraci dati su na slikama 26, 27 i 28. Na prethodno opisan način ručno se simulira proces oporavka od katastrofe.

```
eonrokkrl123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances promote-replica testreplica
Promoting a read replica stops replication and converts the instance
to a standalone primary instance with read and write capabilities.
This can't be undone. To avoid loss of data, before promoting the
replica, you should verify that the replica has applied all
transactions received from the primary.

Learn more:
https://cloud.google.com/sql/docs/mysql/replication/manage-replicas#promote-replica

Do you want to continue (Y/n)? Y

Promoting Cloud SQL replica...
eonrokkrl123@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch testreplica --activation-policy ALWAYS
The following message will be used for the patch API method.
{"name": "testreplica", "project": "bustling-cosmos-318123", "settings": {"activationPolicy": "ALWAYS"}}
Patching Cloud SQL instance...done.

Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/testreplica].
```

Slika 26: Unapređivanje replike za čitanje


```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances describe testreplica
backendType: SECOND_GEN
connectionName: bustling-cosmos-318123:us-central1:testreplica
databaseVersion: MYSQL_8_0
etag: 78d53422ff18dd355af0c0c2889b7b5abda264a2edb0be84358c122641a12c6c
gceZone: us-central1-f
instanceType: CLOUD_SQL_INSTANCE
ipAddresses:
- ipAddress: 34.132.55.238
  type: PRIMARY
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances describe mysql1135
...
tier: db-custom-4-26624
state: STOPPED

```

Sada je primarna instanca!

Slika 27: Provera stanja stare i nove primarne instance

```

eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances patch testreplica --availability-type REGIONAL --enable-bin-log --backup-start-time=02:00
The following message will be used for the patch API method.
{"name": "testreplica", "project": "bustling-cosmos-318123", "settings": {"availabilityType": "REGIONAL", "backupConfiguration": {"backupRetentionSettings": {"retainedBackups": 7, "retentionUnit": "COUNT"}, "binaryLogEnabled": true, "enabled": true, "startTime": "02:00", "transactionLogRetentionDays": 7}}}
Patching Cloud SQL instance...done.

Updated [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/testreplica].
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances create drreplica --master-instance-name=testreplica --region=us-east4
Creating Cloud SQL instance...done.

Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/drreplica].
NAME      DATABASE_VERSION  LOCATION  TIER      PRIMARY_ADDRESS  PRIVATE_ADDRESS  STATUS
drreplica  MYSQL_8_0         us-east4-b  db-custom-4-26624  34.150.217.180   -                RUNNABLE
eonrokkrl23@cloudshell:~ (bustling-cosmos-318123)$ gcloud sql instances delete mysql1135
All of the instance data will be lost when the instance is deleted.
Do you want to continue (Y/n)? Y
Deleting Cloud SQL instance...done.
Deleted [https://sqladmin.googleapis.com/sql/v1beta4/projects/bustling-cosmos-318123/instances/mysql1135].

```

HA konfiguracija

Kreiranje međuregionalne replike za čitanje

Brisanje originalne primarne instance

Slika 28: Podešavanje nove primarne instance i brisanje stare

Ovim je kompletiran ručno izveden process oporavka od katastrofe. Nakon njega, iako je baza podataka u primarnom regionu bila nedostupna, ponovo je uspostavljen sistem poput onog pre pada, koji funkcioniše ispravno.

4. Zaključak

Google Cloud SQL pomaže u izradi relacionih baza podataka za aplikacije u oblaku i njima u potpunosti upravlja. Nudi mogućnost stvaranja baza podataka koristeći MySQL, PostgreSQL i SQLServer. Cloud SQL je DbaaS, pa omogućava korisnicima pristup i upotrebu sistema baze podataka u oblaku bez kupovine i postavljanja sopstvenog hardvera, instaliranja softvera baze podataka ili samog upravljanja bazom podataka.

Omogućava skaliranje, visoku dostupnost, replikaciju, kreiranje automatskih rezervnih kopija, oporavak od grešaka i jednostavnu integraciju sa ostalim Google Cloud servisima. Cloud SQL automatski osigurava pouzdanost, sigurnost i skalabilnost baza podataka, tako da korisnici mogu neometano da ih koriste. Automatizovano je kreiranje rezervnih kopija, replikacija, enkripcija i povećanje kapaciteta, istovremeno obezbeđujući dostupnost veću od 99,95%, bilo gde u svetu.

Upravljanje Cloud SQL instancama obavlja se preko Google Cloud platforme ili Google Cloud komandne linije i identično je za sva tri tipa podržanih baza, što je olakšavajuće za korisnika jer ne zahteva poznavanje konkretnih komandi, već Cloud SQL konkretne komande izvršava u pozadini. Zbog toga korisnici mogu više da se fokusiraju na same podatke, umesto na upravljanje bazom podataka.

5. Literatura

1. Rami Honing, (2019), *What is Database-as-a-Service*, <https://www.stratoscale.com/blog/dbaas/what-is-database-as-a-service/>
2. Yifat Perry, (2021), *Google Cloud Database Services Explained*, <https://cloud.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads>, (jun 2021)
3. Google, (2021), *Cloud SQL: Key terms*, <https://cloud.google.com/sql/docs/key-terms>, (jun 2021)
4. Google, (2021), *Cloud SDK: Command Line Interface*, <https://cloud.google.com/sdk/gcloud/reference>, (jun 2021)
5. Google, (2021), *Cloud SQL: All features*, <https://cloud.google.com/sql#all-features>, (jun 2021)
6. Google, (2021), *Cloud SQL Documentation*, <https://cloud.google.com/sql/docs>, (jun 2021)
7. Google Cloud Architecture Center, (2021), *Cloud SQL database disaster recovery*, <https://cloud.google.com/architecture/intro-to-cloud-sql-disaster-recovery>, (jun 2021)