UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

TEMA: REPLIKACIJA MYSQL BAZE PODATAKA (REPLICATION AND MIRRORING)

Seminarski rad

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

|  |  |
| --- | --- |
| Student: | Mentor: |
| Ivan Šušter, br. ind. 1548 | Doc. dr Aleksandar Stanimirović |

Niš, Mart 2024. godina

SADRŽAJ

[1. UVOD 1](#_Toc164114886)

[2. REPLIKACIJA BAZE PODATAKA 2](#_Toc164114887)

[2.1. Tipovi replikacija u MySQL bazi podataka 2](#_Toc164114888)

[2.2. Razlika između mirroring-a i replikacija 4](#_Toc164114889)

[3. KONFIGURISANJE MySQL REPLIKACIJE 5](#_Toc164114890)

[3.1. Tradicionalna replikacija u MySQL-u pomoću log fajla 5](#_Toc164114891)

[3.1.1. Podešavanje izvornog (master) servera 5](#_Toc164114892)

[3.1.2. Podešavanje replike (slave) servera 7](#_Toc164114893)

[3.1.3. Testiranje master i slave servera 9](#_Toc164114894)

[3.2. Replikacija u MySQL-u pomoću GTID 10](#_Toc164114895)

[3.2.1. Podešavanje master servera 10](#_Toc164114896)

[3.2.2. Podešavanje slave servera 11](#_Toc164114897)

[3.2.3. Testiranje master i slave servera 12](#_Toc164114898)

[4. KOMANDE ZA RAD SA MySQL REPLIKACIJAMA 14](#_Toc164114899)

[5. GRUPNE REPLIKACIJE U MySQL 18](#_Toc164114900)

[6. ZAKLJUČAK 19](#_Toc164114901)

[LITERATURA 20](#_Toc164114902)

# 1. UVOD

U kontekstu baza podataka, važno je održati efikasnost, doslednost i otpornost na greške kako bi podaci bili stabilni i lako dostupni. Replikacija baze podataka ima značajnu ulogu u ovom procesu, jer omogućava distribuciju i usklađivanje podataka preko više servera, poboljšavajući tako dostupnost i čitanje podataka.

Usredsređujući se na MySQL replikaciju, jedan od vodećih sistema za upravljanje bazama podataka, predstavljeni su osnovni principi replikacije, sa pregledom različitih tipova replikacije dostupnih unutar MySQL sistema. Zatim sledi analiza procesa konfigurisanja MySQL replikacije, uz korake za postavljanje tradicionalne replikacije pomoću binarnih log datoteka, obuhvatajući podešavanje primarnog (ili master) i replike (ili slave) servera.

Dodatno, predstavljena je upotreba Global Transaction Identifiers (GTID) za replikaciju, naglašavajući prednosti ovog pristupa. Nadalje, pružen je pregled korisnih komandi za upravljanje MySQL replikacijama i objašnjen je koncept grupne replikacije. U pitanju je moderna funkcionalnost unutar MySQL-a koja pruža visok nivo dostupnosti i otpornosti na greške u distribuiranim okruženjima.

# 2. REPLIKACIJA BAZE PODATAKA

Replikacija baze podataka uključuje kreiranje i održavanje kopija objekata baze podataka na više servera u sistemu distribuirane baze podataka. Replikacija je osnovni mehanizam koji poboljšava skalabilnost i toleranciju grešaka u bazama podataka. Omogućava da podaci budu dostupni sa različitih servera, omogućavajući paralelnu obradu upita i balansiranje opterećenja [1].

Postoje različiti tipovi mehanizma replikacije baze podataka, kao što su replikacija *master-slave,* jednosmerna replikacija podataka, *full* replikacija, delimična replikacija i skalabilna replikacija za transakcione web aplikacije. Ovi mehanizmi služe u različite svrhe kao što su pravljenje rezervnih kopija, izveštavanje, distribucija podataka i obezbeđivanje dostupnosti i pouzdanosti podataka [2].

Na osnovu prethodnih definicija može se zaključiti da je replikacija baze podataka veoma bitan koncept u sistemima za upravljanje bazama podataka, koji nudi prednosti kao što su skalabilnost, otpornost na otkaz servera itd. i zbog toga postoje različiti mehanizmi odnosno tipovi replikacije da bi se zadovoljili različiti zahtevi i obezbedio integritet podataka u distribuiranim okruženjima.

Još jedna definicija replikacije bi bila da ona predstavlja proces kopiranja baze podataka sa jednog servera na jedan ili više drugih servera kako bi se poboljšala dostupnost baze podataka, kao i sakalabilnost i performase baze podataka. Ovaj proces takođe obuhvata i sinhronizaciju “glavne”baze i replika te glavne baze [3].

## 2.1. Tipovi replikacija u MySQL bazi podataka

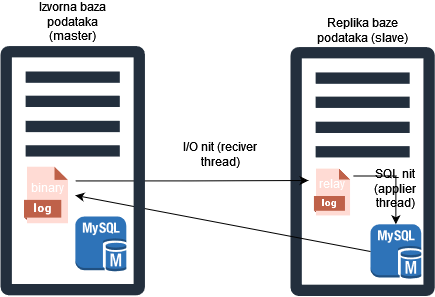
U MySQL-u replikacija je asinhrona, replike ne moraju biti povezane stalno da bi dobijale nove promene od izvorne ili glavne baze i u zavisnosti od konfiguracije moguće je praviti replike svih baza, odabranih baza ili samo odabanih tabela iz neke baze podataka.

Neke od prednosti koje pruža MySQL replikacija su [4]:

* Skaliranje - raspoređivanje opterećenja između više replika radi poboljšanja performansi. U ovom okruženju, sva upisivanja i ažuriranja se moraju obavljati na izvornom odnosno master serveru. Čitanje iz baze se mogu obavljati i na nekoj od replika. Ovaj model može poboljšati performanse upisivanja (zato što je izvorni server posvećen samo upisivanju i ažuriranju), dok veoma povećava brzinu čitanja sa povećanjem broja replika.
* Sigurnost podataka - pošto replika može pauzirati proces replikacije, moguće je pokrenuti proces pravljenja rezervne kopije na replici bez konflikta sa postojećim podacima na izvornom serveru.
* Analitika - bez ometanja i uticaja na performanse izvornog servera se mogu dodavati podaci na izvornom serveru, dok se vrši analiza informacija na replici.
* Distribucija podataka na daljinu - može se koristiti replikacija za kreiranje lokalne kopije podataka koje će koristiti udaljene lokacije bez trajnog pristupa izvornom serveru.

U MySQL-u postoje dve vrste replikacija i to:

1. Tradicionalni metod - obuhvata upotrebu binarnog log fajla za replikaciju promena napravljenih na glavnom (*master*) serveru na jedan ili više *slave* servera. Ovaj metod se oslanja na evidentiranje promena u binarnom log fajlu, a zatim ponavljanje ovih promena na ostalim serverima da bi sinhronizovao podatke.
2. Replikacija globalnim identifikatorima transakcija (engl. *Replication with Global Transaction Identifiers - GTID*) *-* je naprednija i novija metoda koja obezbeđuje jedinstveni identifikator za svaku transakciju na svim serverima u topologiji replikacije. GTID pojedonstavljuje proces praćenja i upravljanja replikacijom, posebno u složenim postavkama replikacije sa više servera. Dodeljivanje GTID-a pravi razliku između transakcija na izvoru i repliciranih transakcija koje se reprodukuju na replici. Kada se transakcija izvrši na izvornom serveru dodeljuje joj se novi GTID, pod uslovom da je transakcija upisana u binarni log fajl.

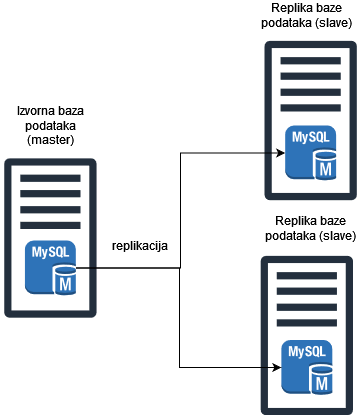


Slika 1 Replikacija korišćenjem binarnog log fajla

Ove dve metode nude različite pristupe replikacije baze podataka. Tradicionalni metod korišćenja log fajla je dobro uspostavljena metoda koja se široko koristi za replikaciju promena podatka između servera. Nasuprot tome, replikacija sa GTID-om uvodi efikasniji način upravljanja replikacijom.

Korisnici MySQL-a mogu odabrati pristup koji najbolje odgovara njihovim potrebama replikacije, bilo da je to tradicionalna metoda ili GTID za poboljšano upravljanje i praćenje transakcija na distribuiranim serverima.

MySQL takođe podržava različite tipove sinhronizacije prilikom replikacije. Osnovni tip sinhronizacije je jednosmerna (engl. *one-way*)*,* asinhrona replikacija, gde jedan server služi kao glavni ili izvorni server, dok jedan ili više drugih servera služe kao replike. I ovakav vid replikacije u MySQL-u je najkorišćeniji.



Slika 2 Jednosmerna (one-way) replikacija

Pored asinhronih u MySQL-u postoje i sinhrone replikacije koje su karakteristične za NDB klaster. Kao i polu-sinhrone (engl. *semisynchronous*) i odložene (engl. *delayed*) replikacije

U nastavku rada biće dat primer konfiguracije MySQL-a za replikaciju korišćenjem binarnog log fajla.

## 2.2. Razlika između mirroring-a i replikacija

Dok replikacija predstavlja proces kopiranja podataka sa izvornog servera na servere gde se nalaze replike, *mirroring* predstavlja identičnu kopiju baze podataka. Pojam *mirroring* označava da promene koje su nastale u bazi se istog trenutka preslikavaju i primenjuju na kopijama ili replikama.

U MySQL-u kod *master-slave* replikacije glavni odnosno izvorni server obrađuje operacije upisivanja i ažuriranja i prenosi promene na jednu ili više replika asinhrono i ovakva konfiguracija pruža prednosti kao što je raspodela opterećenja (load balancing), ona ne garantuje doslednost podataka u realnom vremenu između glavne baze i replika.

Sa druge strane preslikavnje (mirroring) daje prioritet konzistentnosti podataka u realnom vremenu. To znači da kada se izvrši promena u glavnoj bazi, ona se odmah preslikava na “replike”, obezbeđujući da sve kopije baze podataka ostanu identične u bilo kom trenutku.

U suštini replikacija i preslikavanje dele cilj kopiranja podataka na više servera, preslikavanje daje prioritet sinhronoj replikaciji kako bi podaci ostali identični i dostupni na bilo kom serveru ali takođe i povećava opterećenje mreže. Zbog te mogućnosti predstavlja i bolji način za pravljenje rezervne kopije podataka od asinhrone replikacije.

# 3. KONFIGURISANJE MySQL REPLIKACIJE

U ovom poglavlju biće predstavljeni primeri osnovne konfiguracije replikacija u MySQL-u. U prvom primeru biće korišćen tradicionalni pristup replikacije pomoću binarnog log fajla.

Za primer će biti potrebno dve virtuelne mašine u ovom slučaju operativni sistem u virtuelnim mašinama je Ubuntu Server 22.04, kao i instaliran MySQL sistem za upravljanje bazama podataka, u ovom slučaju verzija 8.

## 3.1. Tradicionalna replikacija u MySQL-u pomoću log fajla

### 3.1.1. Podešavanje izvornog (master) servera

Da bi konfigurisali izvorni ili master server da koristi binarni log fajl za replikaciju potrebno je da omogućimo binarni log i dodeliti serveru jedinstveni ID. Pored toga potrebno je i konfigurisati *firewall* kako bi replika mogla da komunicira sa master serverom to se može uraditi komandom sudo ufw allow from 192.168.1.105 to any port 3306. Važno je napomenuti da svaki server koji se nalazi u topologiji za replikaciju mora imati jedinstveni ID koji je pozitivan ceo broj i koji se postavlja pomoću sistemske promenljive server\_id.

Za postavljanje server\_id promenljive koristi se sledeća komanda u MySQL-u:

SET GLOBAL server\_id = 1;

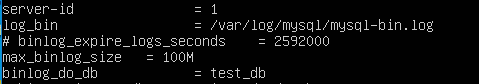
Takođe potrebno je podesiti naziv binarnog log fajla. Ova podešavanja je moguće i uraditi izmenom konfiguracionog fajla koji se nalazi na lokaciji /etc/mysql/mysql.conf.d/mysqld.cnf i potrebno je izmeniti sledeće linije:

bind-adress - upisuje se adresa izvornog servera, određuje adresu sa koje će MySQL dozvoliti konkcije. Na ovoj liniji umesto 127.0.0.1 upisati 0.0.0.0.

server-id - kao što je već navedeno jedinstveno identifikuje server.

log\_bin - putanja do binarnog log fajla kao i naziv binarnog log fajla.

binlog\_do\_db - upisuje se naziv baze podataka koja će biti replikovana. Ako je potrebno praviti replike više baza podataka dodaje još linija sa nazivima baza ispod ove linije. Takođe može se i specificirati naziv baze za koju neće biti pravljena replika korišćenjem direktive binlog\_ignore\_db.

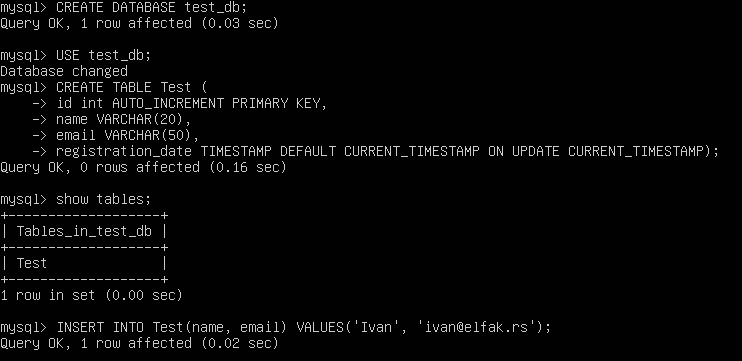


Slika 3 Konfiguracioni fajl MySQL-a na master serveru

Kada su sve promene upisane potrebno je sačuvati konfiguracioni fajl i restartovati mysql servis sledećom komandom.

sudo systemctl restart mysql

Kada se MySQL server ponovo porekene moguće je i logovanje u MySQL pomoću komande mysql -u root -p ukoliko želimo da pristupimo MySQL-u kao *root* korisnik. Dalje je potrebno kreirati bazu podataka sa nazivom koji je upisan u konfiguracioni fajl ranije (Slika 3).



Slika 4 Kreiranje baze podataka i tabele na master serveru

Nakon kreiranja baze potrebno je izvršiti pomoću komande USE test\_db; biramo bazu sa kojom želimo da radimo, odnosno gde će biti kreirana tabela i dodat novi red u tabeli kao na slici 4.

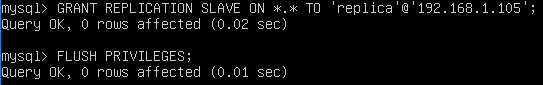
Sada je potrebno kreirati korisnika za replikaciju odnosno korisnika koji će replikacija koristiti da prati promene na master serveru.



Slika 5 Kreiranja korisnika replica na master serveru

Komandom sa slike 5. se kreira korisnik *replica* u ovom slučaju se koristi *IDENTIFIED WITH mysql\_native\_password* za autentikaciju korisnika, podrazumevani mehanizam autentikacije bi zahtevao korišćenje enkriptovane konkcije između izvornog servera i replike. Ova opcija bi trebalo da se koristi u produkciji.

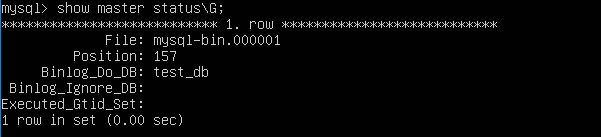
Kada je korisnik kreiran potrebno mu je dodeliti privilegije pomoću komande prikazane na slici 6.



Slika 6 Dodeljivanje privilegija korisniku replica na masteru

Nakon čega je potrebno uraditi *FLUSH* kako bi se oslobodio keš koji je upisan prilikom izvršavanja komande za kreiranje korisnika i dodeljivanje privilegije korisniku. Kako bi dobili koordinate binarnog fajla odnosno pozicije za podešavanje slave servera, potrebno je zaključati tabele u bazi komandom FLUSH TABLES WITH READ LOCK; ovo takođe omogućava ukoliko je potrebno tj. ukoliko baza već sadrži podatke da se napravi *dump* fajl sa podacima ovo je moguće uraditi komandom sudo mysqldump -u root test\_db > test\_db.sql. Sada je moguće ponovo otključati tabele komandom UNLOCK TABLES;, a kreirani dump fajl je moguće kopirati na slave server komandom scp test\_db.sql ivan@192.168.1.105:/tmp/.

Nakon toga proces podešavanja master servera je završen, sada je moguće pokrenuti komandu kao na slici 7. gde se dobija prikaz statusa master servera.

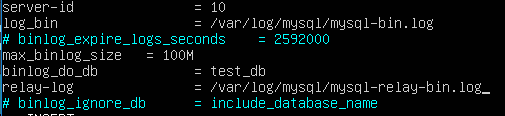


Slika 7 Status master servera

Sa slike 7. se može zaključiti da je fajl u kome se upisuje binarni log *mysql-bin.000001* i pozicija 157, ove dve infromacije su binte za podešavanje replike ondnosno *slave* servera.

### 3.1.2. Podešavanje replike (slave) servera

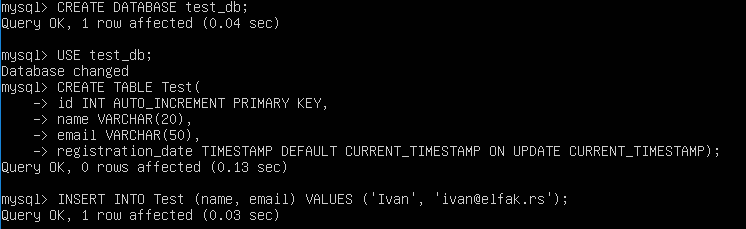
Kada odradimo podešavanja na masteru potrebno je podesiti i repliku ili više njih ukoliko je potrebno. Kao prva stavka prilikom podešavanja replike jeste izmena konfiguracionog fajla koji se nalazi na istoj lokiaciji kao i kod izvornog servera tj. /etc/mysql/mysql.conf.d/mysqld.cnf, gde je potrebno izmeniti linije u fajlu kao na slici ispod (Slika 8).



Slika 8 Konfiguracioni fajl MySQL baze podataka na replika (slave) serveru

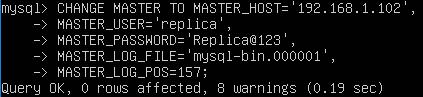
Bitno je uneti server-id koji nije isti kao kod master servera, kao i dodati novu liniju *relay-log* koja označava lokaciju i naziv relay log fajla na serveru. Još jedna bitna napomena da je potrebno obrisati *auto.cnf* fajl koji se nalazi na lokaciji /var/lib/mysql/auto.cnf razlog brisanja ovog fajla je taj što služi za dodeljivanje UUID-a automatski, a s obzirom da je virtuelna mašina klonirana sa instaliranim MySQL serverom, javiće se greška prilikom pokretanja slave-a.

Nakon toga potrebno je restartovati MySQL servis i ulgotvati se u MySQL gde je potrebno kreirati bazu podataka i tabelu u njoj kao i na master serveru (Slika 9).



Slika 9 Kreiranje baze, tabele i dodavanje novog reda

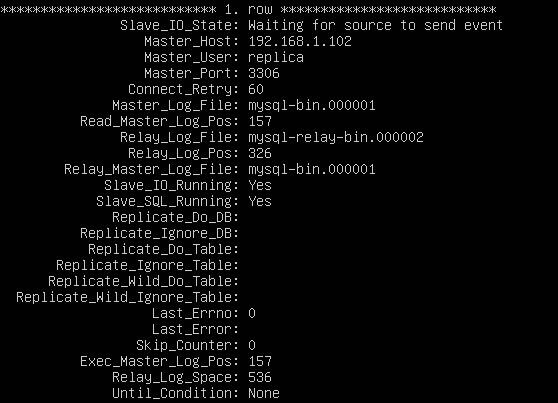
Kada je to sve završeno potrebno je izvršiti komandu kojom se omogućava da se replikacija poveže na master server, komanda je prikazana na slici ispod. Za MASTER\_HOST *-* se unosi IP adresa master servera, MASTER\_USER *-* korisnik koji je kreiran ranije na master serveru, MASTER\_PASSWORD *-* lozinka koja je dodeljena korisniku *replica* koji je kreiran na izvornom serveru, MASTER\_LOG\_FILE *-* log fajl koji će replika pratiti i iz koga preuzima promene nad bazom podataka, a koji je takođe naveden u konfiguracionom fajlu MySQL-a i MASTER\_LOG\_POS *-* pozicija koja je dobijena ranije kada je prikazan status master servera.



Slika 10 Primer podešavanja master-a na slave serveru

Jedna od napomena je da se umesto *master* i *slave* naziva mogu koristiti SOURCE i REPLICA, tako bi primer sa slike 10 mogao da se napiše i kao CHANGE REPLICATION SOURCE TO SOURCE\_HOST itd.

Nakon izvršene prethodne komande, može se izvršiti komda za pokretanje slave servera START SLAVE; i onda se može proveriti status servera pomoću komande SHOW SLAVE STATUS; prikazane na slici 11.



Slika 11 Status slave servera

Iz status može se videti na kojoj IP adresi je master server, koji je korisnik na masteru preko kog pristupamo masteru, da master server radi na portu 3306 koji je podrazumevani port za MySQL, takođe se vidi naziv binarnog log fajla u kome se upisuju pomene, kao i relay log fajl na slave serveru. Pored toga mogu se videti i dve linije Slave\_IO\_Running i Slave\_SQL\_Running koje predstavljaju procese koje se kreiraju na slave serveru i o kojima je bilo reči na početku da IO nit služi za osluškivanje promena u binarnom fajlu na masteru i preno si ih u relay binarni fajl na slave-u, dok SQL nit čita promene iz relay binarnog fajla i izvršava ih na slave serveru. Ovde se takođe mogu videti greške koje se javljaju.



Slika 12 Status slave servera nastavak

Sa slike 12 se može videti da je Slave\_SQL\_Running\_State pročitao sve promene iz relay log fajla i da je spreman za čitanje daljih promena.

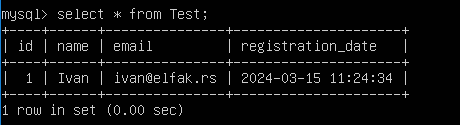
Takođe SQL\_Delay predstavlja vreme posle kog će slave server primeniti promene koje su se desile na masteru, o ovome je isto bilo reči na početku rada, to su odložene replikacije koje se mogu konfigurisati takođe u mysqld.cnf fajlu.

### 

### 3.1.3. Testiranje master i slave servera

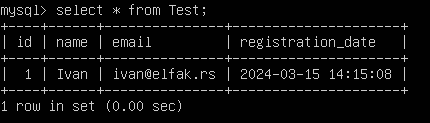
Kada su završena sva podešavanja i kada je provereno da master i slave serveri funkcionišu bez grešaka, može se izvršiti testiranje replikacije odnosno da li se promene preslikavaju sa master servera na slave server.

Prvo se na master serveru unosi komanda za prikaz podataka iz tabele Test kao na slici 13 gde se vidi da trenutno postoji samo jedan red u tabeli.



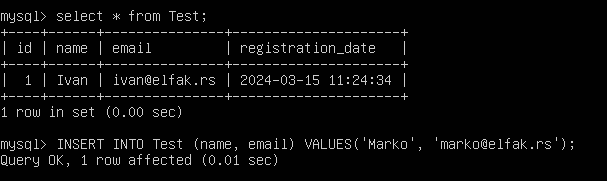
Slika 13 Prikaz podataka iz tabele Test na master serveru

Na slave serveru se takođe unosi komanda za prikaz podataka iz tabele Test koja isto sadrži jedan red. (Slika 14)

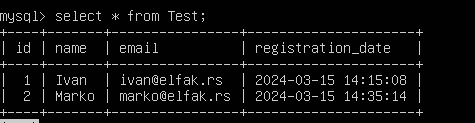


Slika 14 Prikaz podataka iz tabele Test na slave serveru

Sada se na master serveru može dodati još jedan zapis u tabeli Test kao na slici 15. gde se vidi da je dodavanje bilo uspešno, pa nakon toga se može izvršiti provera na slave serveru (Slika 16) gde je prikazana tabela Test i gde se može videti da je red koji je dodat na master serveru preslikan i na slave serveru.



Slika 15 Dodavanje novog reda na master serveru



Slika 16 Prikaz tabele Test na slave serveru nakon dodavanja reda na master serveru

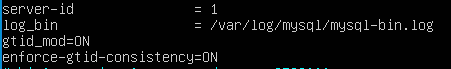
## 3.2. Replikacija u MySQL-u pomoću GTID

Globalni identifikator transakcije (GTID) se kreira i povezuje sa svakom transakcijom koja je uspešno izvršena na izvornom serveru. On obezbeđuje da se svaka transakcija može lako pratiti u sistemu.

U nastavku će biti prikazan primer podešavanja jednostavne *master-slave* replikacije korišćenjem GTID-a.

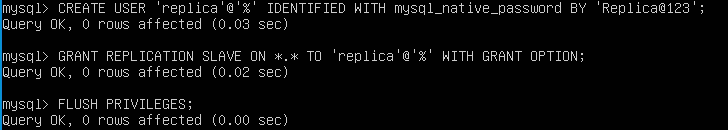
### 3.2.1. Podešavanje master servera

Slično kao kod podešavanja master servera za korišćenje binarnog fajla potrebno je izmeniti konfiguracioni fajl MySQL-a (/etc/mysql-mysql.conf.d/mysqld.cnf). Potrebno je izmeniti bind-address = 0.0.0.0, dodati liniju gtid\_mode=ON i enforce-gtid-consistency=ON. Ostale izmene su prikazane na slici ispod.



Slika 17 Podešavanje master servera za replikaciju korišćenjem GTID-a

Nakon izmene, restartovati MySQL server i dodati pravilo za firewall kako bi replika mogla da komunicira sa bazom pomoću sledeće komande sudo ufw allow from any to any port 3306, zatim otvoriti MySQL prompt gde je potrebno kreirati korisnika sa privilegijama za čitanje promena sa master servera, što je i prikazano na slici 18.



Slika 18 Kreiranje korisnika na master serveru

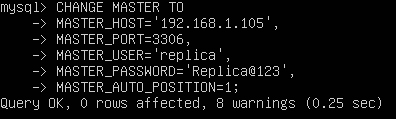
### 3.2.2. Podešavanje slave servera

Na slave serveru potrebno je takođe omogućiti GTID u konfiguracionom fajlu kao na master serveru (potrebno je promeniti i bind-address = 0.0.0.0) i bitno je promeniti server-id da ne bude isti kao kod master servera.



Slika 19 Konfiguracioni fajl na slave serveru

Nakon restarta MySQL servera, u MySQL-u konfigurisati repliku da koristi master sa GTID-om, kao izvor za pravljenje replike, kao i da koristi GTID automatsku poziciju transakcija (Slika 20).

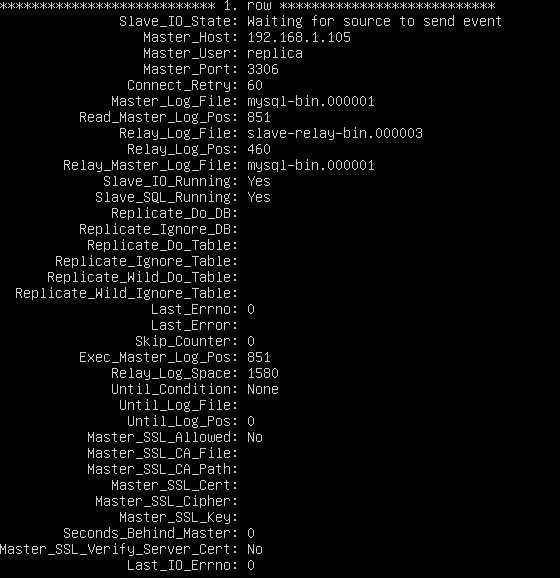


Slika 20 Konfigurisanje replike da koristi master sa GTID-om

Ovakav unos konfiguracije, kada se izostave MASTER\_LOG\_FILE i MASTER\_LOG\_POS znači da je replika konfigurisana da radi sa GTID-om, što označava i omogućavanje automatske pozicije.

Nakon konfiguracije izvršiti pokretanje slave servera komandom, START SLAVE; takođe može se pokrenuti i komanda za proveru statusa slave servera - SHOW SLAVE STATUS\G;

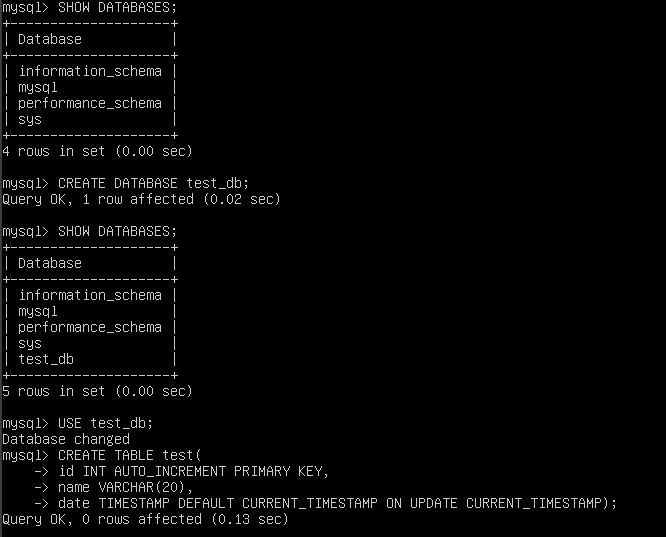
Na slici 21. se može videti status slave servera koji čeka da se desi promena na master serveru kako bi mogao da napravi repliku te promene s obzirom da su IO i SQL niti aktivne i spremene da pročitaju i primene promene na slave serveru.



Slika 21 Slave status

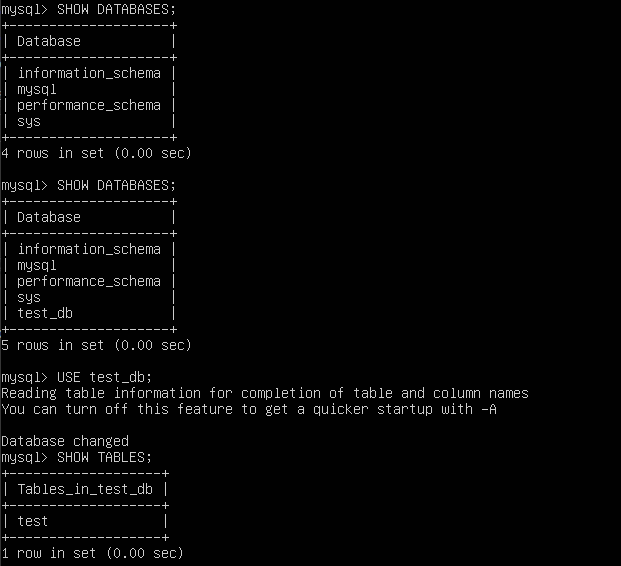
### 3.2.3. Testiranje master i slave servera

Kada je sve podešeno moguće je kreirati bazu na masteru kao na slici 22.



Slika 22 Kreiranje baze i tabele na master serveru

U isto vreme biće kreirana baza i tabela na slave serveru što se može videti na slici ispod.

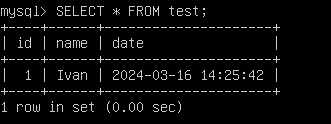


Slika 23 Prikaz stanja baze na slave serveru nakon kreiranja na master serveru

Takođe ukoliko se unese neki red u tabeli koja je kreirana na master serveru, kao na slici 24. biće odmah dostupan i na slave serveru (Slika 25).



Slika 24 Unos reda u tabelu na master serveru



Slika 25 Prikaz tabele na slave serveru nakon unosa podataka na master serveru

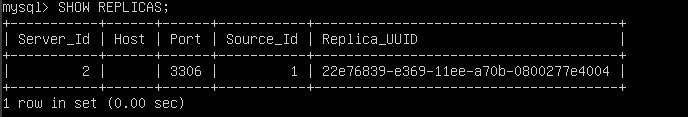
# 4. KOMANDE ZA RAD SA MySQL REPLIKACIJAMA

Nakon kreiranih replika velika je verovatnoća da će biti porebno poznavanje i nekih od komandi koje se koriste za manipulaciju replikacija u MySQL-u. U zavisnosti od vrste potreba i zahteva, važno je zanti dodatne komande koje se mogu iskoristiti u radu sa replikacijama.

Neke od komandi su već prikazane u prehodnom poglavlju prilikom podešavanja replikacija ali je mogće i menjati ih kada god se javi potreba za promenom nekih od podešavanja, tako na primer komanda CHANGE MASTER TO koja se koristi za podešavanje slave servera, se može iskoristiti i kasnije ukoliko je potrebno promeniti neki od parametara.

Pored te komande korišćena je i komanda START SLAVE; koja pokreće slave nakon unosa parametara preko prehodne komande CHANGE MASTER, pored START komande postoje i komanda za stopiranje slave servera STOP SLAVE, kao i za restartovanje odnosno brisanje svih konfiguracija na replikaciji komanda RESET SLAVE ALL. Ove komande se mogu iskoristi u slučaju kada je potrebno na primer unaprediti slave server u master, gde je potrebno redom zaustaviti slave, obrisati binarne log fajlove i izmeniti konfiguraciju.

Na master serveru je takođe moguće izlistati server koji se koriste kao slave serveri kao što je prikazano na slici ispod.

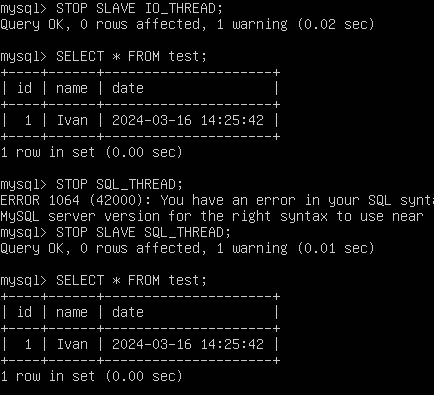


Slika 26 SHOW REPLICAS; komanda za prikaz slave/replica servera

Još neke korisne komande koje se mogu iskoristiti u situacijama kada se pravi rezervna kopija podataka su STOP SLAVE IO\_THREAD čime se stopira čitanje promena iz binarnog log fajla na master serveru. Ovo takođe može biti korisno kada je potrebno da SQL nit dostigne odnosno primeni sve promene koje su pročitane, sa izvornog servera pomoću IO niti, do tog trenutka. Pored toga korisno je i kada je potrebno izvršiti i neke promene na master serveru kao i na replici.

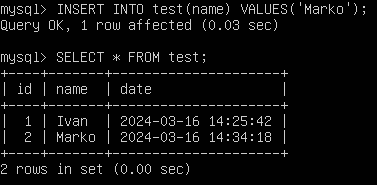
Takođe je moguće stopiranje i SQL niti komandom STOP SLAVE SQL\_THREAD čime se onemogućava izvršenje promena koje je IO nit prenela sa master servera na repliku.

U primeru datom na slici 27. dat je primer stopiranja niti, odnosno stopirane su IO i SQL niti na slave serveru.



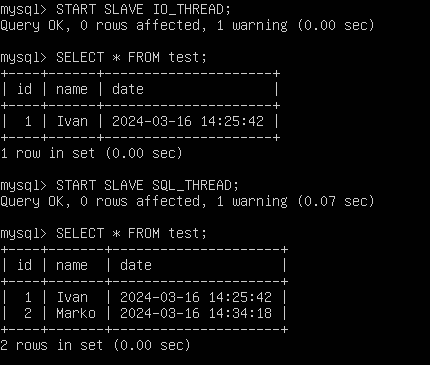
Slika 27 Stopiranje IO niti i SQL niti na slave serveru

Na masteru se dodaje novi red sa podacim u tabeli (Slika 28).



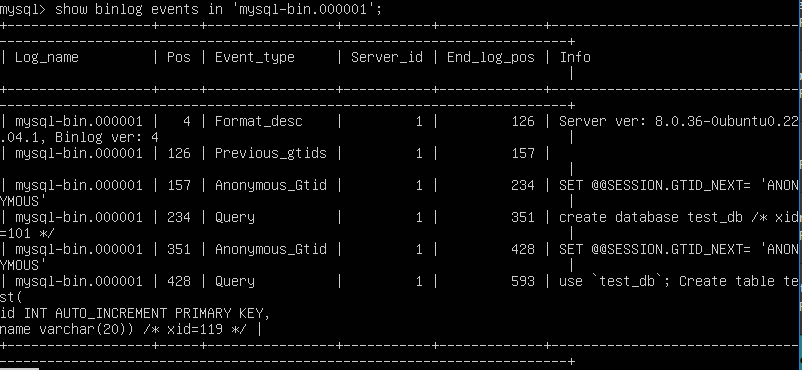
Slika 28 Dodavanje novog reda na master serveru

Na sledećoj slici je prikazano startovanje prvo IO niti, pa provera tabele gde se može videti da još uvek ništa nije promenjeno na slave serveru. Nakon toga je startovana i SQL niti koja izvršava promene koje je IO nit kopirala sa master servera na slave server.



Slika 29 Startovanje IO i SQL niti na slave serveru

Za pregled binarnih fajlova se mogu koristit sledeće dve komande, prva je za pregled binarnog log fajla SHOW BINLOG EVENTS IN ‘mysql-bin.000001’; koja se može izvršiti na master serveru gde se dobija rezultat kao na slici ispod.



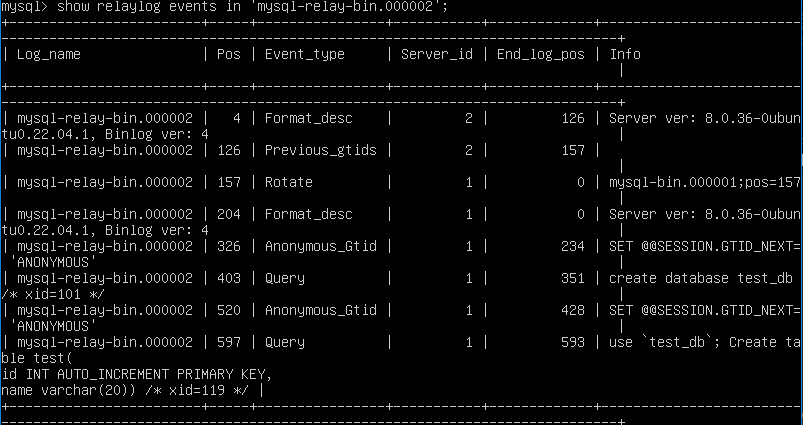
Slika 30 Prikaz binarnog log fajla na master serveru

Gde se može videti naziv log fajla, pozicija, tip promene koji se desio na serveru odnsno u bazi, identifikacioni broj servera na kome se desila promena kao i kraj pozicije promene. Umesto ove komande, moguće je i prikazivanje binarnog fajla pomoću komande sudo mysqlbinlog /var/log/mysql/mysql-bin.000001 gde se mogu videti detalji binarnog fajla kao što je npr. vreme kada je izvršena promena, takođe se može videti thread\_id koja nit je izvršila promenu, exec\_time vreme za koje je izvršena promena itd. (Slika 31).



Slika 31 Prikaz binarnog log fajla pomoću komande mysqlbinlog

Na slave serveru je moguće prikazati promene u relay log fajlu takođe pomoću komande mysqlbinlog s obzirom da ova dva log fajla imaju isti format ili je moguće korišćenje komande SHOW RELAYLOG EVENTS IN ‘mysql-relay-bin.000002’; kao što je prikazano na slici 32. format je isti kao i kod binarnog log fajla na master serveru.



Slika 32 Prikaz promena u relaybin log fajlu na slave serveru

Još jedna komanda koju je moguće iskoristi u slučaju da replikacija prestane sa radom pa je potrenbo oporaviti replikaciju je komanda za preskaknje promena koje su se desile na master serveru i ona se može izvršiti samo dok su niti na slave serveru stopirane u suprotnom se javlja greška SET GLOBAL sql\_slave\_skip\_counter=2, čime se preskaču sledeće 2 promene koje su se dogodile na master serveru.

Ove komande omgoćuvaju administratorima da manipulišu replikama od konfigurisanja replika do nadgledanja satusa replikacije i rešavanje problema. Efikasno korišćenje ovih komandi je ključno za održavanje pouzdanosti i efikasnosti replikacija u MySQL-u.

# 5. GRUPNE REPLIKACIJE U MySQL

MySQL grupna replikacija je funkcija u MySQL-u koja omogućava da više servera rade zajedno, replicirajući podatke na različitim čvorovima kako bi se osigurala visoka dostupnost i tolerancija grešaka. Ovaj funkcija omogućava grupi MySQL servera da koordinišu i dogovore promene koje se dešavaju u bazi podataka obezbeđujući da svaki server ima identičnu kopiju podataka. Ovo podešavanje pruža prednosti kao što je automatsko rešavanje grešaka, u slučaju da jedan server otkaže, drugi server u grupi može preuzeti da bi se održao kontinuitet dostpnosti.

Glavne karakteristike grupne replikacije uključuju:

* Transakcije se repliciraju sinhrono da bi se održala doslednost podataka na svim članovima u grupi.
* Grupna replikacija osigurava detekciju i oporavak od grešaka i na taj način osigurava dostupnost baze.
* Svaki server u grupi može izvršavati i operacije upisivanja i operacije čitanja što omogućava balansiranje opterećenja i skalabilnost.
* Novi serveri se mogu dodavati odnosno pridružiti grupi dinamički, kao i serveri članovi se mogu izbaciti iz grupe.

Još jedna bitna stvar je kako se izvršavaju promene u grupnoj replikaciji tj. kada je transakcija čitanja i pisanja spremna za *commit,* izvorni server (na kome je izvršena promena) emituje promene i njihove identifikatore svim serverima u grupi. Ovakvo emitovanje promena je atomsko što znači da ili svi serveri dozvole promene ili ih odbace [5].

MySQL grupna replikacija i tradicionalna replikacija se razlikuju u nekoliko ključnih aspekata. Tradicionalna replikacija u MySQL-u obično uključuje konfiguraciju master-slave gde jedan server odnosno master replicira podatke na jedan ili više slave servera. Ova replikacija je asinhrona, što znači da može doći do kašnjenja u replikaciji podataka između master i slave servera. S druge strane, MySQl grupna replikacija uključuje više servera koji rade zajedno odnosno sinhrono kako bi kopirali podatke na svim čvorovima istovremeno.

# 6. ZAKLJUČAK

Danas, replikacija baze podataka je nezaobilazna komponenta u arhitekturi baza podataka, ključna za održavanje visoke dostupnosti, skalabilnosti i otpornosti sistema na greške.

Replikacija MySQL baze podataka omogućava distribuciju podataka preko višestrukih servera, čime se poboljšavaju performanse čitanja i generalna efikasnost aplikacija, omogućavajući brži pristup podacima sa različitih lokacija.Takodje, replikacija baze podataka nudi rešenja i strategije za oporavak podataka u slučaju kvara sistema, čime se značajno smanjuje rizik od gubitka informacija. Zahvaljujući njoj, sistemi su sposobni da odgovore na porast zahteva za podacima, optimizuju raspodelu opterećenja i unaprede korisničko iskustvo.

# LITERATURA

[1] B. Kemme and G. Alonso, "Database replication", Proceedings of the VLDB Endowment, vol. 3, no. 1-2, p. 5-12, 2010. https://doi.org/10.14778/1920841.1920847

[2] S. Yadav, G. Singh, & D. Yadav, "Mathematical framework for a novel database replication algorithm", International Journal of Modern Education and Computer Science, vol. 5, no. 10, p. 1-10, 2013. https://doi.org/10.5815/ijmecs.2013.09.01

[3] https://bs.linux-console.net/?p=3997#gsc.tab=0 (Pregledano: 25.2.2024.)

[4] https://dev.mysql.com/doc/refman/8.0/en/replication.html (Pregledano: 27.2.2024.)

[5] https://dev.mysql.com/doc/refman/8.0/en/group-replication-background.html (Pregledano: 1.3.2024.)