**Univerzitet u Nišu**

**Elektronski fakultet**

MySQL - replikacija baze podataka

Predmet: Sistemi za upravljanje bazama podataka

Student: Profesor:

Nevena Čolić 1360 prof. dr. Aleksandar Stanimirović

# Sadržaj

[MySQL - opšte](#_ow0suu78ye3o) 3

[Replikacija baza podataka](#_zg8v07apcvel) 4

[Replikacija MySQL baze podataka](#_6rtqyjrldp1k) 7

[Korišćenje replikacije radi čuvanja rezervne kopije](#_r0h3hl2onzl7) 17

# MySQL - opšte

MySQL je sistem za upravljanje bazom podataka razvijen od strane Oracle-a. U pitanju je relaciona baza podataka koja čuva podatke u odvojenim tabelama radije nego u jednom velikom skladištu.

Strukture baze su organizovane u fajlove optimizovane za postizanje veće brzine. Za fleksibilno programsko okruženje koristi se logički model sa objektima kao što su tabele, pogledi, redovi i kolone. Korisnik definiše pravila o vezama između različitih entiteta, kao što je jedan-pre-jedan, jedan-prema-više, da li je polje jedinstveno, obavezno kao i pokazivače između različitih tabela.

MySQL je otvorenog koda, što znači da bilo ko može da koristi i modifikuje softver. Dostupan je svakome i besplatan. MySQL koristi GPL[[1]](#footnote-0) za definisanje pravila korišćenja softvera u različitim situacijama ali i razne licence za komercijalno korišćenje. MySQL server se može koristiti na personalnim računarima ali i na umreženom klasteru. U pitanju je klijent/server sistem koji se sastoji od višenitnog SQL servera sa podrškom za različite backend-ove, klijentske programe i biblioteke, administrativne alate ali i ugrađenom bibliotekom za rad sa nitima.

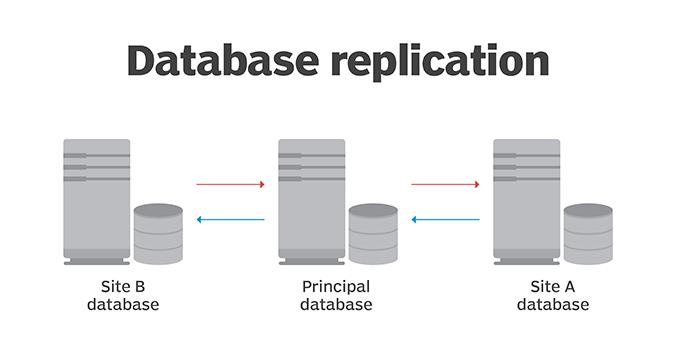
MySQL je napisan u C i C++ jezicima i radi na Windows, Linux i macOS operativnim sistemima. Dizajniran je da bude potpuno višenitan i da po potrebi koristi nekoliko CPU-a ako su oni dostupni. Koristi veoma brze tabele zasnovane na B stablu i brz sistem za alokaciju memorije koji koristi više niti. Implementira memorijske heš tablice koje se koriste kao privremene tabele. Prilikom obrade upita, najčešće nema nikakve alokacije memorije nakon inicijalizacije upita zbog veoma brzih SQL funkcija.



**Slika 1. MySQL logo**

# Replikacija baza podataka

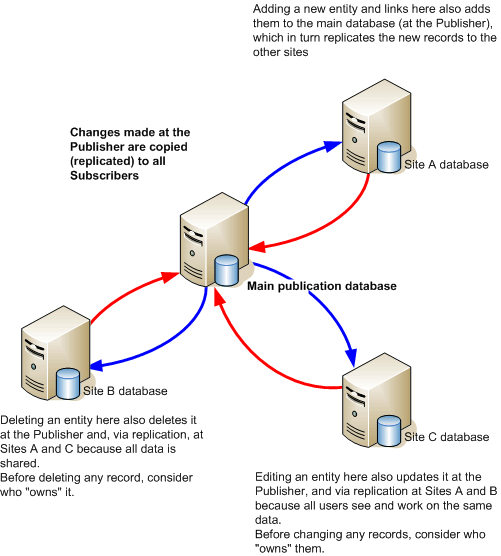
Replikacija baze podataka podrazumeva učestalo kopiranje podataka iz baze jednog servera u bazu drugog, tako da svi korisnici dele iste informacije. Kao rezultat se dobija distribuirana baza podataka u kojoj korisnici mogu na brz način pristupiti relevantnim podacima bez uticaja na druge korisnike.



**Slika 2. Grafički prikaz primera replikacije**

Replikacija može biti proces koji se odvija jednom ili stalno i uključuje sve izvore podataka u distribuiranoj strukturi jedne organizacije. DDBMS[[2]](#footnote-1) je zadužen da se dodavanja, brisanja ili bilo kakve promene nad podacima odmah preslikaju nad podacima na svim lokacijama. U pitanju je infrastruktura koja omogućava ili i sama izvršava replikaciju baze podataka.

Centralna baza podataka se naziva bazom publikacija (eng. publication database) zbog toga što pruža podatke korisnicima. Podaci iz ove baze se kopiraju (replikuju) do baza koje su za to pretplaćene (eng. subscribers) i nalaze se na drugim lokacijama. Promene nastale nad jednim subscriberom se kopiraju do baze izdavača (eng. publisher) a on zatim kopira tu promenu dalje. Na Slici 2. prikazana je glavna baza konfigurisana kao izdavač i povezana sa tri pretplaćene baze.



**Slika 2. Baza Publisher i tri baze Subscriber**

Tehnike replikacije

**Asinhrona -** podaci se šalju posebnom, model serveru. Model server je server koji uzima replike podataka od klijenta. On zatim kontaktira klijenta sa potvrdom da su podaci primljeni. Podaci se dalje kopiraju do drugih replika.

**Sinhrona -** podaci se šalju model serveru i zatim se replikuju svim replika serverima pre nego što se klijent obavesti da su podaci uspešno replikovani. Ovaj metod traje duže od asinhronog ali pruža prednost saznavanja da su svi podaci uspešno replikovani pre daljeg rada.

Asinhrona replikacija nudi fleksibilnost i lakoću korišćenja, s obzirom da se replikacija dešava u pozadini. Međutim, postoji rizik gubitka podataka bez znanja klijenta s obzirom da potvrda stiže pre početka procesa replikacije. Sinhrona replikacija je stroža i oduzima više vremena ali pruža veće šanse sigurnosti da će proces biti uspešno izvršen. U slučaju neuspeha, klijent će biti obavešten jer potvrda dolazi tek po okončanju čitavog procesa.

Po tipu arhitekture, razlikuju se sledeće (model se odnosi na čvora vođu):

* Arhitektura sa jednim modelom (eng. single-leader architecture) - jedan server koji prima upise od klijenta i podaci replika se i dobijaju sa istog. Ovo je najčešći metod iako je sinhron.
* Arhitekture sa više modela (eng. multi-leader architecture) - više servera prima zahteve upisa i služe kao model replikama.
* Arhitekture bez modela (eng. no-leader architecture) - svaki server koji može primati upise ali i služiti kao model replikama. Ovo je prvi put predstavljeno od strane Amazona kod DynamoDB-a. Iako pruža maksimalnu fleksibilnost, suočava se sa izazovima kod sinhronizacije.

| PREDNOSTI | MANE |
| --- | --- |
| **Manje opterećenje -** replikovani podaci se mogu podeliti na više servera i na taj način rasporediti opterećenje | **Gubitak podataka -** može se desiti sa netačnim podacima ili kada se baza ažurira i dogodi se hazard zbog brisanja koje se u međuvremenu dogodilo |
| **Efikasnost -** serveri sa manje upita nude poboljšane performanse manjem broju korisnika | **Nedoslednost podataka -** slično, kada dođe do hazarda nad podacima |
| **Visoka dostupnost -** više servera koji čuvaju iste podatke su zaslužni za visoku dostupnost u slučaju otkaza nekog od servera | **Manje opterećenje -** rad više servera zahteva i veću cenu |

# Replikacija MySQL baze podataka

Replikacija omogućava kopiranje podataka iz MySQL baze jednog servera tj. u ovom slučaju izvora (eng. source) u jedan ili više MySQL servera koji se nazivaju replike. Podrazumevana konfiguracija jeste asinhrona replikacija, tačnije, replike ne moraju stalno biti povezane kako bi primale obaveštenja od izvora. Zavisno od konfiguracije, moguća je replikacija celih baza, samo odabranih baza ili čak izabranih tabela iz baza. Prednosti replikacije kod MySQL-a uključuju:

* Rešenja za skaliranje - raspoređivanje opterećenja između nekoliko replika poboljšava performanse. U ovakvom okruženju, svi upisi i ažuriranja moraju biti izvršeni na izvoru. Čitanja se međutim, mogu izvršiti na jednoj ili više replika. Ovaj model poboljšava performanse upisa (pošto je izvorni server posvećen ažuriranjima), dok se brzina čitanja drastično povećava sa većim brojem replika.
* Bezbednost podataka - pošto replika može pauzirati proces replikacije, moguće je napraviti rezervnu kopiju podataka replike bez oštećivanja odgovarajućih izvornih podatka.
* Analitika - podaci koji se dobijaju u realnom vremenu se mogu kreirati na izvoru, dok se analitika informacija može odvijati na replici bez bilo kakvog uticaja na izvorni server.
* Distribucija podataka na većoj udaljenosti - replikacija se može koristiti za kreiranje lokalne kopije podataka za korišćenje od strane nekog udaljenoj sajta, bez stalnog pristupa izvornom serveru.

U MySQL-u, replikacija podrazumeva da izvorna baza zapisuje sve promene nad podacima u specijalni fajl pod nazivom binarni log. Jednom kada je replika inicijalizovana, kreira procese koji su podeljeni na po dve niti. Prva, pod nazivom IO nit, se povezuje na izvornu MySQL instancu i čita binarni log, red po red i zatim sve to kopira u lokalni fajl pod nazivom prenosni log (eng. relay log). Druga nit, zvana SQL nit, čita akcije iz relay loga i zatim ih primenjuje na instanci replike što je brže moguće.

Novije verzije MySQL-a podržavaju dva načina replikovanja podataka. Razlika je u načinu na koji replike vode računa o tome koje su akcije već obradile.

Prvi način je tradicionalni i poznat je pod nazivom replikacija zasnovana na poziciji unutar binarnog loga (eng. binary log file position-based replication). Kada je instanca MySQL servera pretvori u repliku, neophodno joj je dati skup koordinata za binarni log. On se sastoji iz imena binarnog log fajla na izvornom serveru koji replika mora pročitati i

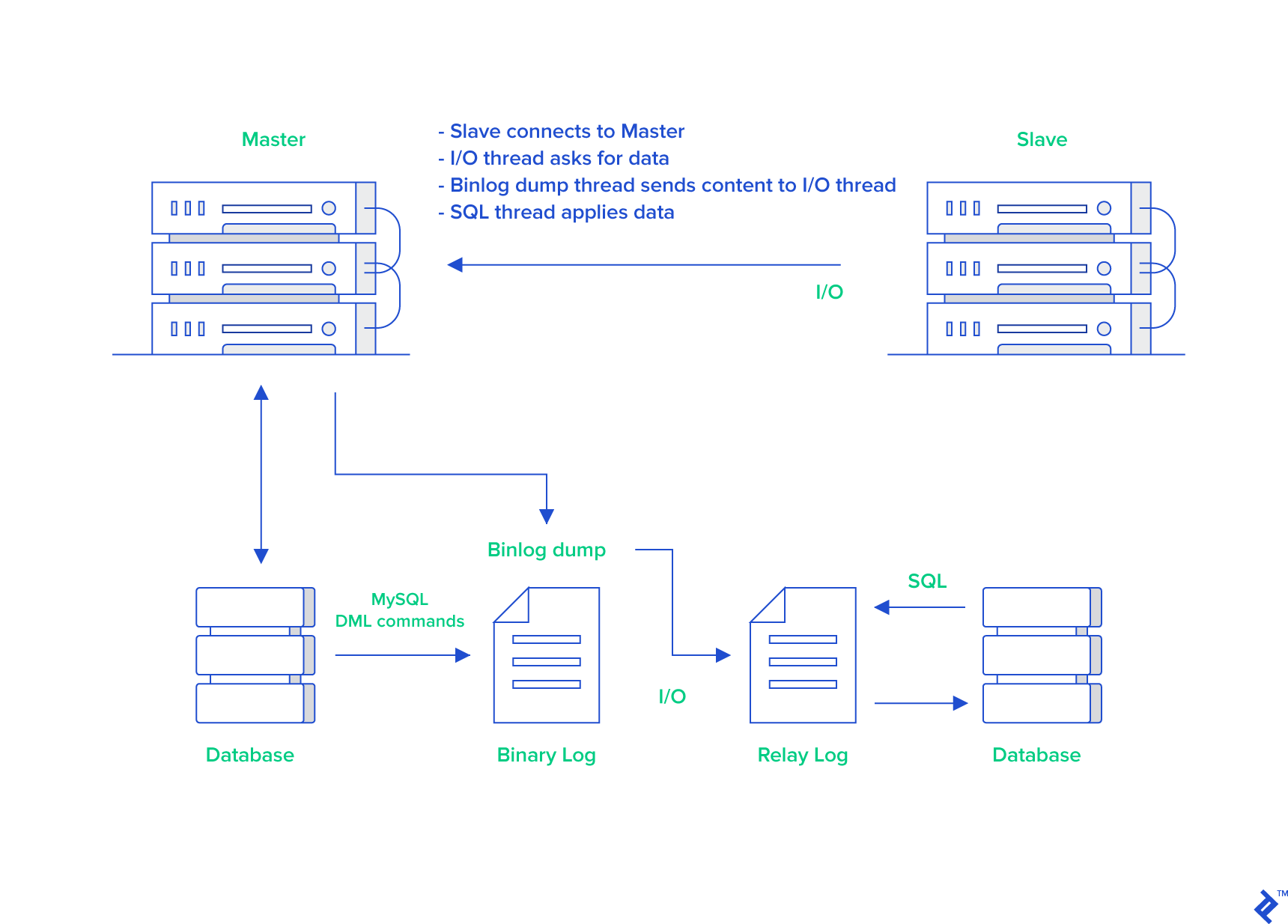
odgovarajuće pozicije unutar fajla koja predstavlja bazu koju bi replika trebalo kopirati. Ove koordinate su bitne jer bi replika bez njih počela replikovati svaku naredbu izvršenu nad bazom ikada. Ovo može dalje dovesti do problema ako je traženo da se kopiraju podaci počev od određene tačke u vremenu.

Noviji metod je zasnovan na transakcijama (eng. transaction-based replication) i uključuje kreiranje globalnog identifikatora transakcije (eng. GTID) za svaku transakciju. Mehanika iza ove vrste replikacije je veoma slična prethodnoj: kada god se dogodi bazna transakcija na izvoru, MySQL joj dodeljuje i zapisuje GTID za datu transakciju u binarni log zajedno sa samom transakcijom. GTID i transakcija se zatim šalju do replika za dalju obradu. Ovaj metod ima brojne prednosti. Na primer, zbog toga što i izvor i replike čuvaju GTID-eve koje su obradili, ukoliko se bilo ko susretne sa GTID-em koji je već obrađen, ta transakcija će biti preskočena. Replike takođe ne moraju više da znaju koordinate unutar binarnog log fajla da bi mogle da obrade transakciju. Sve ovo vodi do toga da je startovanje novih replika i promena redosleda postojećih u lancu replikacije dosta manje komplikovano.

Replikacija u MySQL-u podržava takođe i nekoliko tipova sinhronizacije. Originalni tip je jednosmera, asinhrona replikacija u kojoj se jedan server ponaša kao izvor dok su ostali replike. Ovo predstavlja kontrast sinhronoj replikacije koja je karakteristična za NDB Cluster[[3]](#footnote-2). U najnovijoj verziji MySQL-a, podržana je i polusinhrona replikacija. Kada je u pitanju polusinhrona replikacija, izvršena akcija na izvoru se blokira pre povratka sesiji koja je prvobitno i izvršila transakciju dok najmanje jedna replika ne javi da je primila i logovala događaj za transakciju. MySQL 8.0 takođe podržava i odloženu replikaciju tako da replika namerno kasni za izvorom za neko unapred definisano vreme. U slučaju da je neophodno koristiti sinhronu replikaciju, preporuka je koristiti NDB Cluster.

Master-Slave replikacija

Ovaj vid replikacije omogućava administratoru baze da replikuje ili kopira podatke koji se čuvaju, jednom ili više servera odjednom. Na ovaj način uvek postoji rezervna kopija svih podataka. U nekim slučajevima, kada master server ne radi, odmah se može preći na neku od replika bez uticaja na krajnjeg korisnika. U master-slave replikaciji, podaci se čuvaju prvo na master serveru i zatim se kopiraju dalje. Operacija upisa se vrši samo nad master bazom dok se čitanje može obavljati bilo gde. Replike sluge (eng. slave) se koriste takođe i da smanje i izbalansiraju opterećenje master servera. Svrha master-slave replikacije jeste da sve vreme postoji rezervna kopija sistema koja će biti unapređena u mastera ukoliko isti otkaže.



**Slika 3. Grafički prikaz MySQL replikacije**

Implementacija Master-Slave replikacije

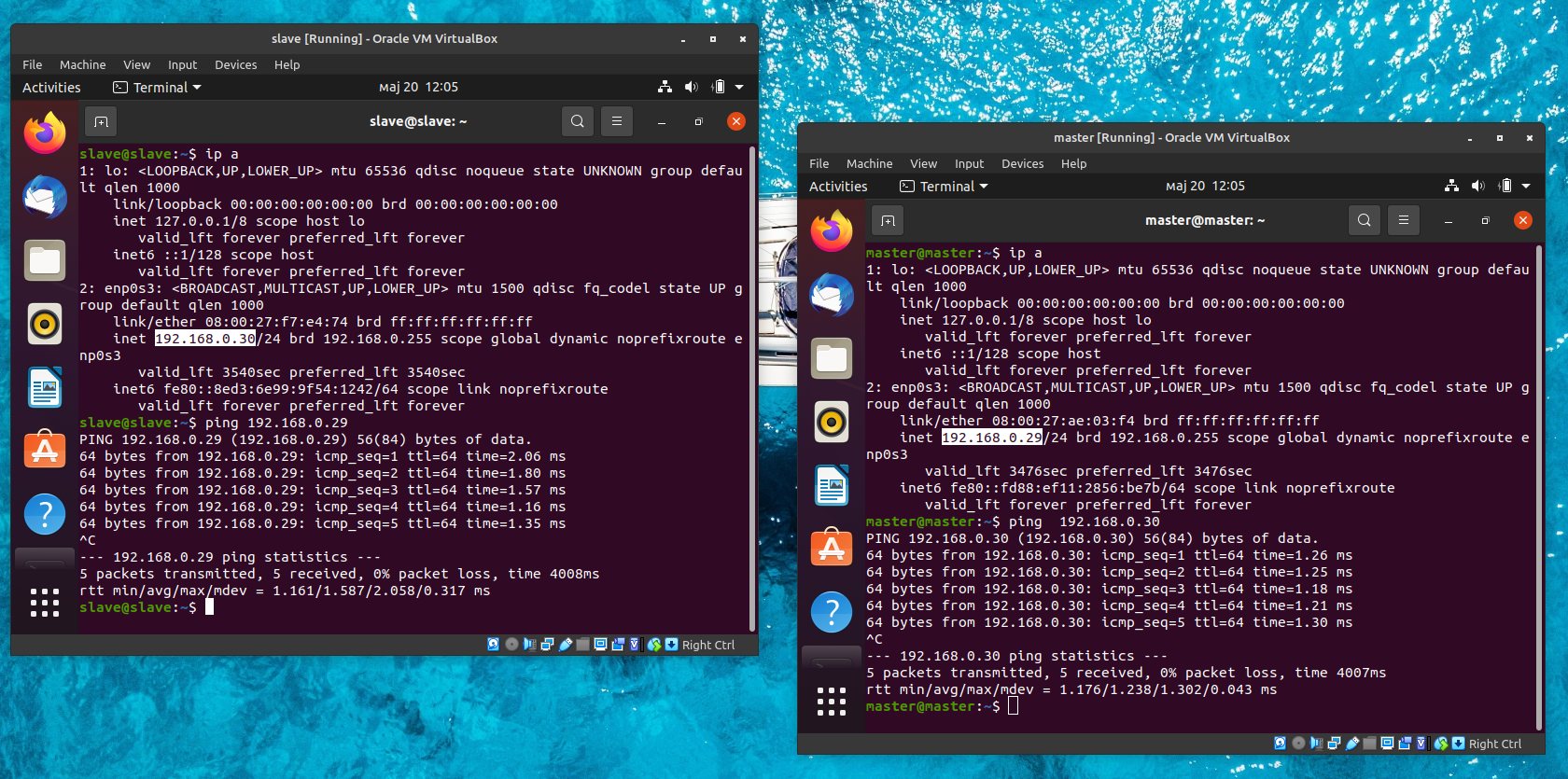
Replikacija je zasnovana na tome da master server vodi računa o svim promenama koje se odvijaju nad bazom (ažuriranja, brisanja i sl.) i zapisuje ih u svoj binarni log. Binarni log, kao što je već objašnjeno, služi kao zapis o svim akcijama koje su se dogodile nad bazom i modifikuju njenu strukturu ili podatke od trenutka kada je ona pokrenuta. SELECT iskazi se ne pamte zbog toga što oni ni na koji način ne utiču na strukturu baze ili njene podatke.

Svaka replika koja se poveže na izvorni, master server zahteva kopiju binarnog loga. Tačnije, ona povlači podatke sa mastera, ne čeka da ih master server sam pošalje. Replika dalje izvršava i sve akcije iz binarnog loga koji je dobila. Rezultat jeste ponavljanje originalnih promena kao što su one odrađene i na master serveru. Kreiraju se tabele ili se menja njihova struktura, podaci se dodaju, brišu i ažuriraju u skladu se akcijama koje se odvijaju na izvornom serveru.

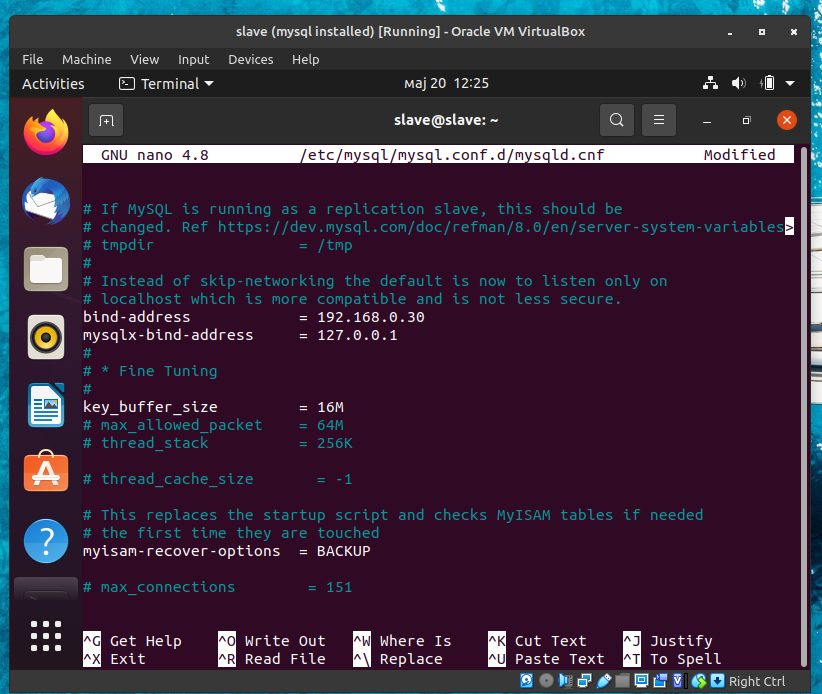
Za primer replikacije potrebno je pokrenuti i konfigurisati dve virtualne mašine gde će jedna biti master a druga slave. Za OS najčešće se koristi Ubuntu 20. Najpre je potrebno podesiti mrežu svakoj od virtualnih mašina na *bridged network (mrežni most).* Most povezuje virtualnu mašinu na mrežu koristeći mrežni adapter hosta. Za svaku od mašina je potrebno proveriti IP adresu i pingovati ih međusobno kako bi se potvrdilo da

je podešavanje prošlo uspešno.

Za najbolje rezultate, preporuka je da obe virtualne mašine rade istovremeno, jedna pored druge, radi lakšeg praćenja celog procesa. Sledeći korak jeste instalacija MySQL-a. Jednom kada je MySQL instaliran potrebno je ažurirati podešavanja obe mašine promenom linija u fajlu **/etc/mysql/mysql.conf.d/mysqld.cnf**. Potrebno je promeniti server id, IP adresu i ime baze koja ce biti replikovana i na kraju restartovati servis.



**Slika 4. Ping mašina**



**Slika 5. Postavljanje IP adrese replike u MySQL konfiguraciji**

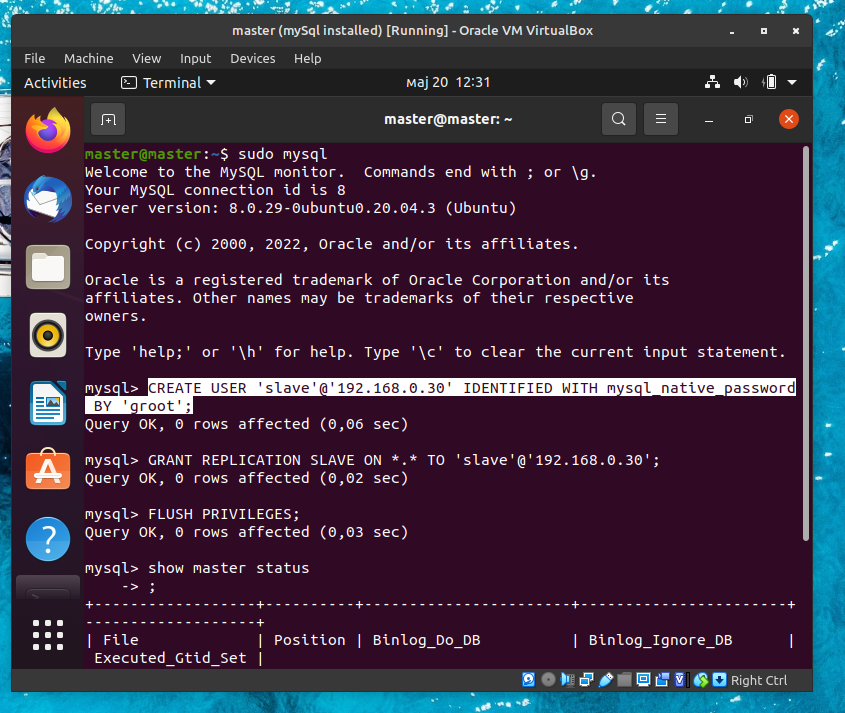
Jednom kada su i master i slave adekvatno podešeni, potrebno je na master serveru kreirati MySQL korisnika koji predstavlja slave server kao što je prikazano na Slici 6, komandom:

*CREATE USER ‘<replicaUserName>@’<replicaIP>’ IDENTIFIED WITH my\_sql\_native\_password BY ‘<somePassword>’;*

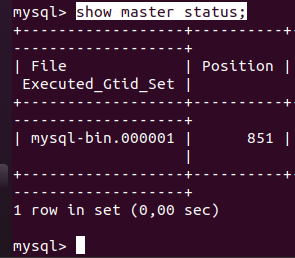
Kreiranom korisniku je potrebno dozvoliti i replikaciju naredbom:

*GRANT REPLICATION SLAVE ON \*.\* TO ‘<imeKreiranogKorisnika>’@’<IPAdresa>’;*

Prateći ove dve komande, najbolje je na kraju uraditi FLUSH PRIVILEGES kako bi se očistio keš koji je pre toga bio sašuvan. Da je sve prošlo uspešno može se proveriti proverom master statusa kao na Slici 7.

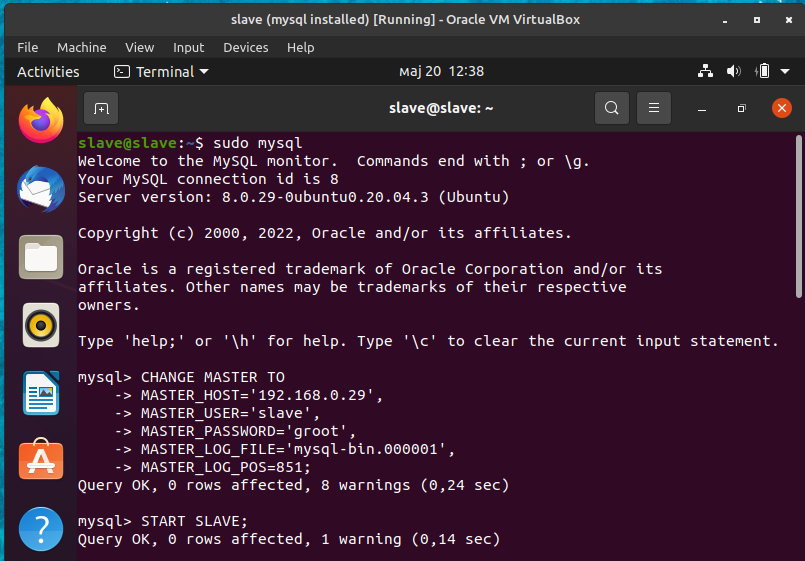


**Slika 6. Kreiranje MySQL slave korisnika na master server**



**Slika 7. Naziv i pozicija master bin log fajla**

# Dalje je potrebno reći replici ko je master server i pokrenuti je:



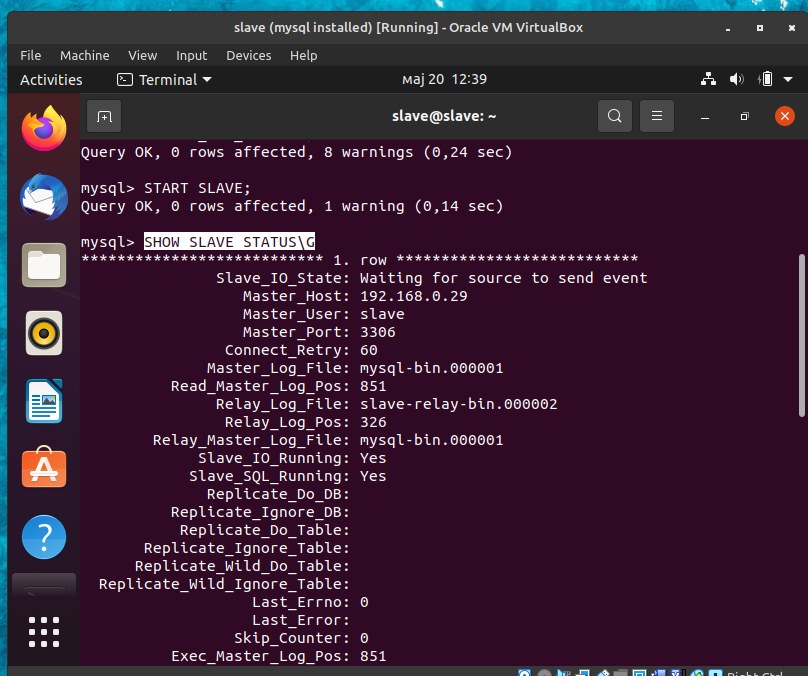
**Slika 8. Podešavanje masters na replici**

Jednom kada replika ima podešenu replikaciju ona će odmah pokušati da se poveže na IP adresu navedenu u konfiguraciji na Slici 8, koristeći korisničko ime i šifru koji su pre toga kreirani na izvornom serveru. Oni se takođe specificiraju replici u ovom koraku kao promenljive *MASTER/SOURCE\_USER* i *MASTER/SOURCE\_PASSWORD*. Replika će takođe tražiti i binarni log fajl sa imenom zadatim u konfiguraciji i počeće da ga čita počev od pozicije nakon *MASTER/SOURCE\_LOG\_POS.*

Za proveru komunikacije između mastera i replike koristi se komanda:

*SHOW SLAVE STATUS\G.*

U najnovijoj verziji MySQL-a se ne preporučuje korišćenje naziva SLAVE već REPLICA, međutim sve naredbe se i dalje isto izvršavaju.

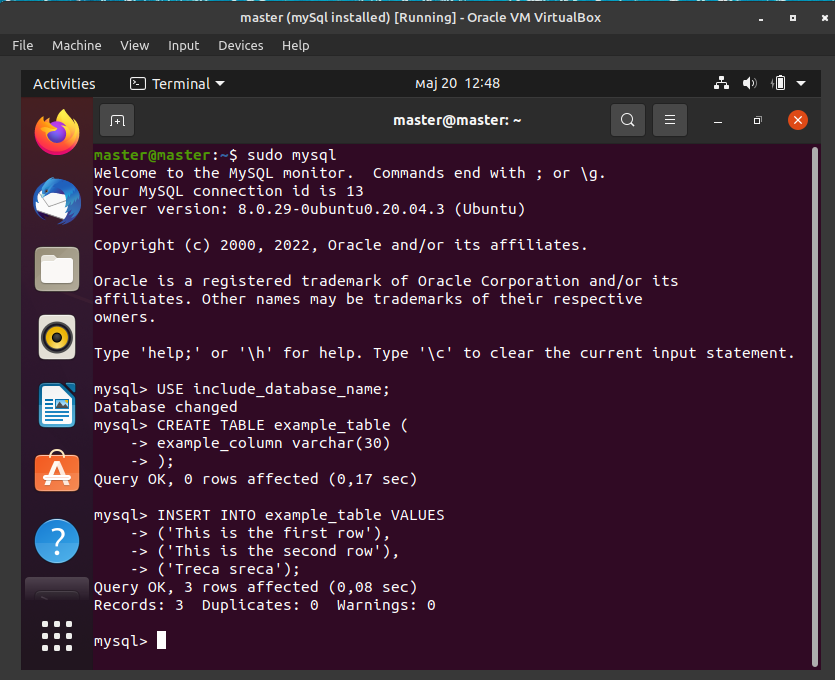


**Slika 9. Prikaz stanja replike**

Nakon što je komunikacija uspešno uspostavljena, može se otpočeti proces prebacivanja podataka. Ukoliko izvorni server nema podataka za migriranje ili je u pitanju nova instalacija, u ovom trenutku se mogu otključati tabele naredbom

*UNLOCK TABLES;*

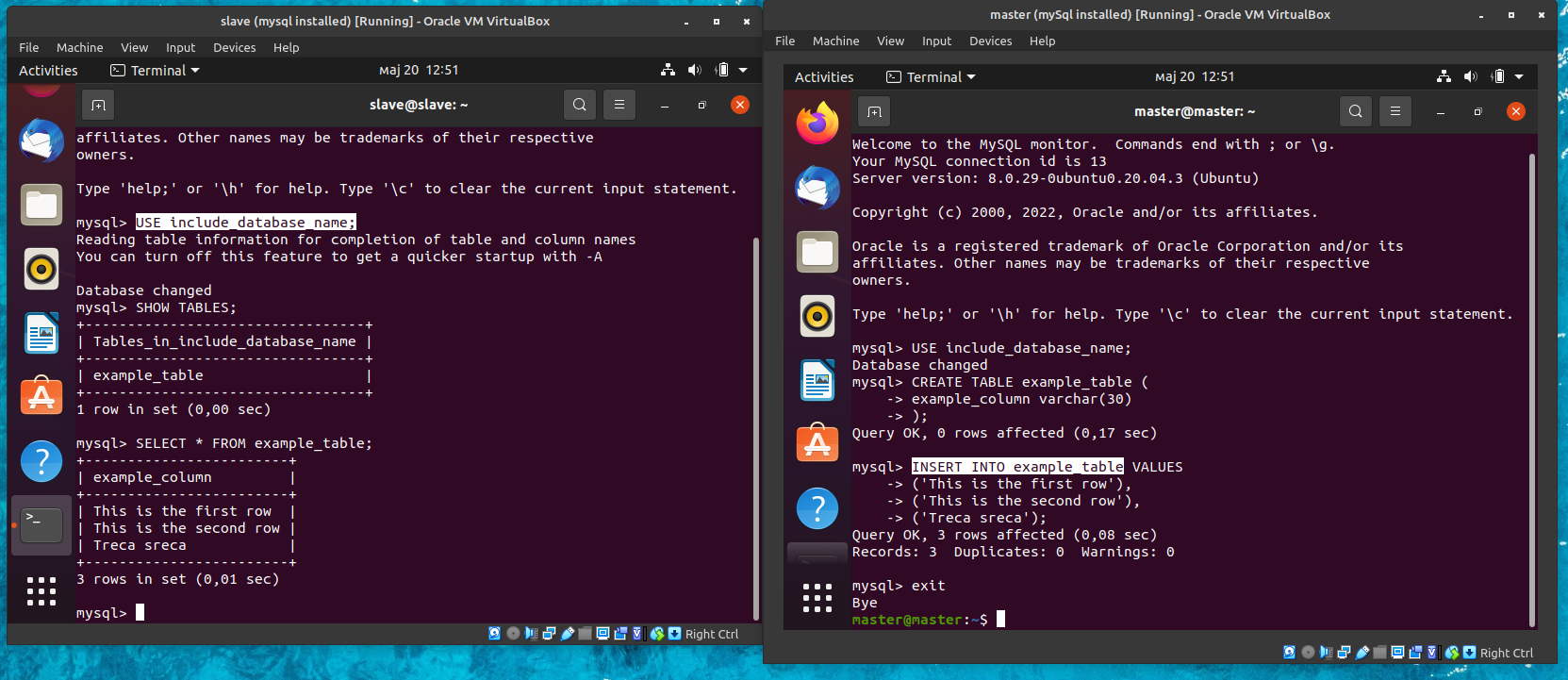
Primera radi, koristiće se baza pod nazivom *include\_database\_name*, sa tabelom *example\_table* i par primerskih podataka. Primer je prikazan na Slici 10.



**Slika 10. Kreiranje master tabela za replikovanje**

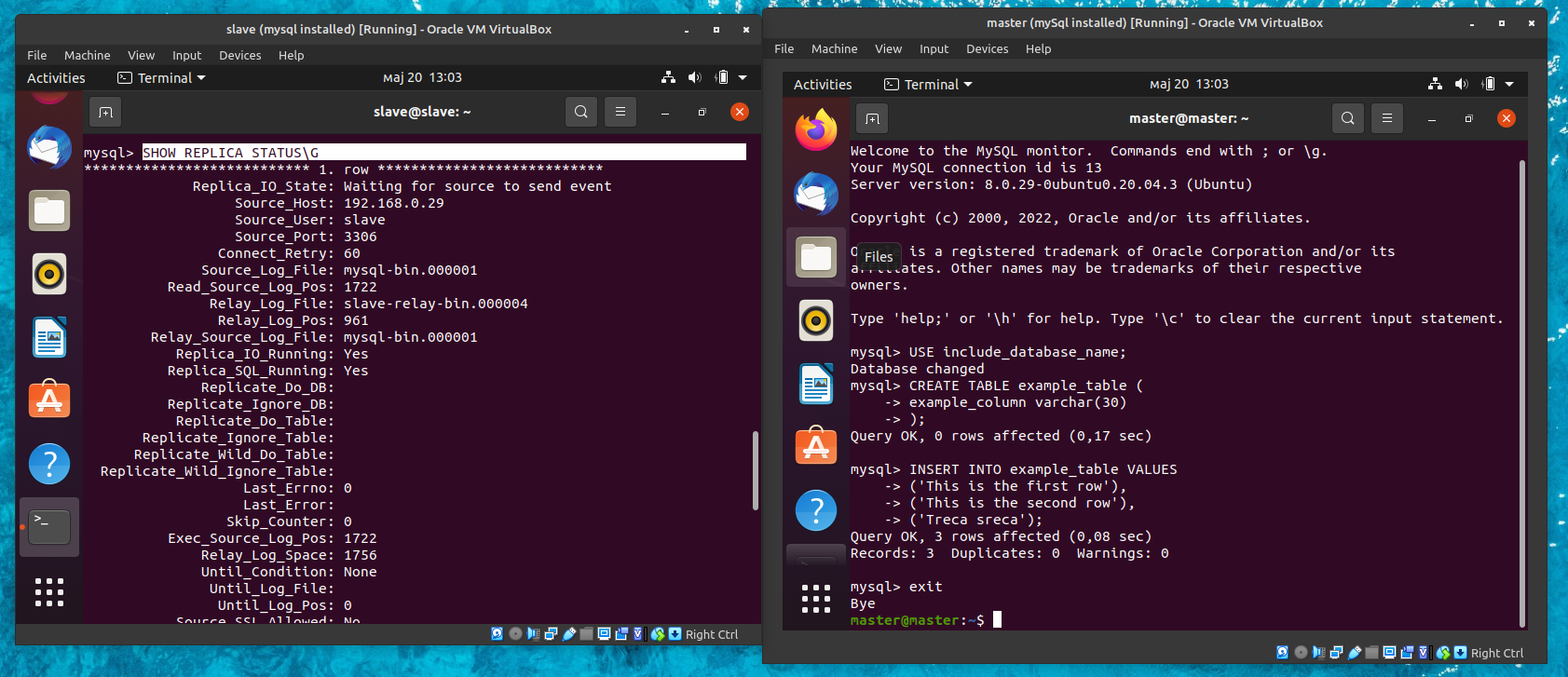
Sve što je sada preostalo jeste prebaciti se na repliku i prikazati tabele kreirane na master serveru, replikacija se već dogodila (videti Sliku 11).

Zbog toga što je svaka replika nezavisna, izvršavanje masterovih akcija se odvija nezavisno na svakoj od replika povezanih sa masterom. Takođe, svaka replika dobija kopiju binarnog loga tako što šalje zahtev izvornom, master serveru i zahvaljujući tome, replika može da čita i ažurira kopiju baze tempom koji njoj odgovara i može započeti i zaustaviti proces replikacije po volji, bez ikakvog uticaja na mogućnost ažuriranja na poslednju, aktuelnu verziju baze ni na master strani niti na strani replike.



**Slika 11. Replikacija master-slave**

Na Slikama 12 i 13. prikazan je status replike i promenljiva koja pokazuje koliko replika kasni za master serverom, što je u ovom slučaju 0.



**Slika 12. Status replike**

# **Slika 13. Prikaz kašnjenja replike za master serverom**

# SHOW SLAVE/REPLICA STATUS iskaz

* Replica\_IO\_State - govori šta nit radi, da li pokušava da se poveže sa masterom (eng. master / source) ili čeka na komunikaciju, da li pokušava da obnovi komunikaciju i sl.
* Master\_Host - Izvorni host na koji je replika povezana.
* Master\_User - korisničko ime naloga koji se koristi za povezivanje sa masterom.
* Seconds\_Behind\_Master - indikator koji specificira koliko replika kasni.
* Master\_Retry\_Count - broj pokušaja replike za konekcijom sa izvornim serverom u slučaju gubitka konekcije.

# Korišćenje replikacije radi čuvanja rezervne kopije

Da bi se replikacija koristila u ove svrhe, potrebno je replikovati podatke sa izvora na replike i zatim sačuvati stanje replika. Replika se može pauzirati i isključiti bez bilo kakvog uticaja na master server, tako da je moguće kreiranje efikasnog “snapshot-a”.

Kako se kreira rezervna kopija podataka zavisi i od veličine same baze kao i od činjenice da li je potrebna rezervna kopija samo podataka ili i stanja replike radi njenog ponovnog kreiranja u slučaju otkaza. Imajući to u vidu, MySQL nudi dve opcije:

* Ukoliko se replikacija koristi kao rešenje za kreiranje rezervne kopije podataka sa izvornog servera i veličina baze nije prevelika, alat mysqldump može biti prikladan.
* Za veće baze podataka, gde mysqldump nije dovoljno efikasan, mogu se sačuvati fajlovi sirovih podataka. Korišćenje ovih fajova znači da se mogu sačuvati rezervne kopije binarnih i relay fajlova i zatim rekreirati replike u slučaju otkaza.

# 

# Literatura

* https://dev.mysql.com/doc/refman/8.0/en/replication-implementation.html
* https://medium.com/@bala529/mysql-master-slave-setup-practice-676e1fa5d627
* https://www.digitalocean.com/community/tutorials/how-to-set-up-replication-in-mysql#step-5-%E2%80%94-configuring-the-replica-database
* <https://hevodata.com/learn/mysql-master-slave-replication/>
* <https://blog.monyog.com/how-to-monitor-mysql-replication/>
* <https://dev.mysql.com/doc/refman/8.0/en/replication-solutions-backups.html>
* <https://dev.mysql.com/doc/refman/8.0/en/replication.html>
* https://www.educative.io/edpresso/what-is-mysql-replication

1. GPL - GNU General Public License [↑](#footnote-ref-0)
2. DDBMS - Distributed Database Management System [↑](#footnote-ref-1)
3. NDM Cluster - Verzija MySQL-a visoke dostupnosti i redundancy prilagođena za distribuirano okruženje [↑](#footnote-ref-2)