

|  |  |
| --- | --- |
|  | UNIVERZITET U NIŠU ELEKTRONSKI FAKULTET |

**Neo4j – Backup/Restore baze podataka**

Seminarski rad

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

|  |  |
| --- | --- |
| Student: | Profesor: |
| Nikola Mitić, br. ind. 1433 | Doc. dr Aleksandar Stanimirović |
|  |  |

Niš, april 2022. godina

SADRŽAJ

[1. Uvod 3](#_Toc104973770)

[2. Backup/Restore u NoSQL bazama podataka 3](#_Toc104973771)

[3. Backup/Restore u Neo4j-u 4](#_Toc104973772)

[3.1. Rezervna kopija onlajn baze podataka 5](#_Toc104973773)

[3.2. Rezervna kopija oflajn baze podataka 10](#_Toc104973774)

[3.3. Oporavak baze podataka 11](#_Toc104973775)

[4. Zaključak 13](#_Toc104973776)

[5. Literatura 14](#_Toc104973777)

# Uvod

Korisnici sa iskustvom u radu i održavanju različitih servera razumeju istinu: „IT sistemi ne mogu dugo da održavaju stabilnost od 100%.“. Kada dođe do kvara sistema, ono što se očekuje je da se brzo popravi i oporavi, ali nažalost, neki kvarovi se ne mogu rešiti glatko ili čak ni u očekivanom vremenskom periodu. Iz ovoga proizilazi da je rezervna kopija toliko važna da može garantovati da se sistem može vratiti u normalno stanje preko postojeće rezervne kopije kada dođe do kvara, što znači da se može izbeći ogroman gubitak zbog nepopravljivog. U nastavku će biti malo više reči o tome kako obezbediti rezervnu kopiju i oporavak sistema kod NoSQL baza podataka, konkretno kod Neo4j baze podataka.

# Backup/Restore u NoSQL bazama podataka

Poslednjih godina NoSQL baze podataka u se pojavile kao alternativa RDBMS sistemima gde aplikacije zahtevaju obim i performanse, a stroga svojstva ACID-a nisu uslov. NoSQL baze podataka nude visoku propusnost, podršku za ogromne strukture podataka i kapacitet horizontalnog skaliranja na račun nepodržavanja relacionih podataka, ACID konzistentnosti i standardne SQL sintakse. Zbog svoje jednostavnosti i fleksibilnosti, NoSQL baze podataka su postale veoma popularne među programerima veb aplikacija. Međutim, većina NoSQL baza podataka obezbeđuje samo osnovne mehanizme rezervne kopije i vraćanja, koji omogućavaju oporavak baza podataka od pada, ali ne i uklanjanje neželjenih operacija izazvanih slučajnim ili zlonamernim radnjama. Većina NoSQL baza podataka nudi jednostavne mehanizme oporavka zasnovane na lokalnim evidencijama i snimcima koji podržavaju oporavak podataka kada se server sruši. Oni takođe koriste globalne evidencije koje održavaju konzistentnost podataka u replikama. Ovi mehanizmi su korisni, ali nisu dovoljni da uklone efekat pogrešnih operacija iz stanja sistema, na primer, eksploatacije ranjivosti veb lokacije ili netačne naredbe za ažuriranje od strane administratora baze podataka koji menja ili briše pogrešan dokument. Ako se izvrši neispravna operacija i ošteti bazu podataka, administrator može da vrati stari snimak koji ne uključuje pogrešnu operaciju. Međutim, iako ovo rešenje uklanja pogrešnu operaciju iz baze podataka, ono takođe odbacuje ispravne promene stanja. Još gore, ako se neispravan rad otkrije kasno, količina izgubljenih podataka može biti ogromna. Bolje rešenje je da ručno izvršite komandu, kao što je ažuriranje ili brisanje, koja uklanja posledice pogrešne operacije, ali to je teško i dugotrajno za administratora. Vreme potrebno za oporavak baze podataka je kritično, ako oporavak traje predugo, moglo bi biti nemoguće uspešno oporaviti podatke bez kolateralne štete.

U zavisnosti od konkretne primene i okruženja, važno je dizajnirati odgovarajuću strategiju pravljenja rezervnih kopija i vraćanja baze podataka. Pritom, treba voditi računa o sledećim faktorima prilikom izbora adekvatne strategije:

* Tip okruženja – razvoj, testiranje ili proizvodnja;
* Obim podataka;
* Dostupni sistemski resursi;
* Koja baza podataka je u pitanju;
* Tolerancija zastoja tokom pravljenja rezervnih kopija i vraćanja;
* Tolerancija gubitka podataka u slučaju kvara;
* Tolerancija zastoja u slučaju kvara;
* Učestalost ažuriranja baze podataka;
* Tip metode rezervne kopije i vraćanja (na mreži ili van mreže);
* Koliko rezervnih kopija želite da zadržite;
* Gde će se čuvati rezervne kopije — disk ili udaljeni server, skladište u oblaku, drugi centar podataka, druga lokacija itd.;
* Kako ćete testirati rutine oporavka i koliko često.

# Backup/Restore u Neo4j-u

Dva su razloga zbog kojih je neophodno pravljenje rezervnih kopija Neo4j baze podataka i njihovo bezbedno off-site skladištenje:

* omogućavanje brzog i efikasnog načina za povratak podataka u slučaju kvara, koji može biti u vezi sa hardverom, ljudskom greškom ili prirodnom katastrofom;
* obavljanje rutinske administrativne operacije, kao što je premeštanje baze podataka sa jedne instance na drugu, nadogradnja ili vraćanje prostora.

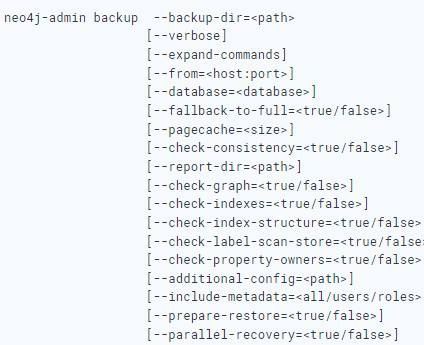
Neo4j podržava pravljenje rezervnih kopija i vraćanje onlajn i oflajn baza podataka. Koristi komande Neo4j Admin alata, koje se mogu pokrenuti iz onlajn, kao i iz oflajn Neo4j DBMS-a. Sve neo4j-admin komande moraju biti pozvane kao neo4j user da bi se osigurale odgovarajuće dozvole za fajl. U nastavku su date neo4j-admin komande:

* *neo4j-admin backup/restore* – koristi se za pravljenje onlajn rezervnih kopija (punih i inkrementalnih) i operacija vraćanja. Baza podataka za pravljenje rezervne kopije mora biti u onlajn režimu. Ova komanda je pogodna za proizvodna okruženja, gde ne možete priuštiti zastoje. Međutim, zahteva više memorije i nije podržan u Neo4j Auri[[1]](#footnote-1);
* *neo4j-admin dump/load* – koristi se za obavljanje oflajn operacija dump-anja i učitavanja. Baza podataka koja ide u dump mora biti u oflajn režimu. Ova komanda je pogodna za okruženja u kojima vreme zastoja nije faktor. Brža je od komande za pravljenje rezervne kopije i proizvodi arhivsku datoteku, koja zauzima manje prostora od normalne strukture baze podataka.
* *neo4j-admin copy* – koristi se za kopiranje oflajn baze podataka ili rezervne kopije. Ova komanda se može koristiti za čišćenje nedoslednosti u bazi podataka, vraćanje neiskorišćenog prostora i migraciju Neo4j 3.5 direktno na bilo koju 4.\* verziju Neo4j-a, uključujući najnoviju verziju, preskačući pritom međukorake.

Kod Neo4j baze podataka je veoma bitno napraviti odvojeno i sigurnosne kopije datoteka neo4j.conf (ako postoji primena klastera, trebalo bi napraviti rezervnu kopiju konfiguracione datoteke za svakog člana klastera), sve datoteke koje se koriste za šifrovanje, tj. privatni ključ, javni sertifikat i sadržaj pouzdanih i opozvanih direktorijuma (ako postoji klaster, trebalo bi napraviti rezervnu kopiju ovih datoteka za svakog člana klastera), i ako se koriste prilagođeni dodaci, potrebno je uveriti se da su oni na bezbednoj lokaciji.

## Rezervna kopija onlajn baze podataka

Kod Neo4j baze podataka rezervna kopija se može napraviti u onlajn režimu korišćenjem backup komande za pravljenje rezervne kopije neo4j-admin-a. Komanda neo4j-admin backup može se koristiti za izvođenje i potpunih i inkrementalnih rezervnih kopija baze podataka na mreži. Potpuna rezervna kopija se vrši kad god nije naveden direktorijum rezervne kopije, najčešće kod prve rezervne kopije na ciljnoj lokaciji. Inkrementalna rezervna kopija se izvodi kad god je naveden postojeći direktorijum rezervne kopije, a evidencije transakcija su prisutne od poslednje rezervne kopije. Komanda backup će zatim kopirati sve nove transakcije iz Neo4j-a i primeniti ih na rezervnu kopiju. Rezultat će biti ažurirana rezervna kopija koja je u skladu sa trenutnim stanjem servera. Komanda backup se može pokrenuti i lokalno i daljinski. Podrazumevano, neo4j-admin backup takođe proverava konzistentnost baze podataka na kraju svake operacije pravljenja rezervne kopije. Međutim, on koristi značajnu količinu resursa, kao što su memorija i CPU. Zbog toga se preporučuje da se rezervna kopija napravi na posebnoj namenskoj mašini. Komanda neo4j-admin backup takođe podržava SSL/TLS. Na slici 1 je prikazana njena sintaksa sintaksa:



*Slika 1: Sintaksa komande neo4j-admin backup za pravljenje onlajn rezervne kopije*

U tablici u nastavku dat je svaki od sufiksa komande neo4j-admin backup kao i njegova namena:

|  |  |
| --- | --- |
| --backup-dir | Ciljni direktorijum |
| --verbose | Omogućavanje detaljanog izlaza |
| --expand-commands | Dozvola za proširenje komande u proceni vrednosti konfiguracije |
| --from | Host i port Neo4j-a |
| --database | Naziv udaljene baze podataka za pravljenje rezervnih kopija.  Vrednost može da sadrži \* i ? za globing, u kojim slučajevima će se napraviti rezervna kopija svih odgovarajućih baza podataka. |
| --fallback-to-full | Ako inkrementalna rezervna kopija ne uspe, rezervna kopija će premestiti staru rezervnu kopiju u <name>.err.<N> i umesto toga koristiti punu rezervnu kopiju. |
| --pagecache | Veličina keša stranice koja se koristi za proces pravljenja rezervnih kopija. |
| --check-consistency | Pokreće proveru doslednosti rezervne kopije baze podataka. |
| --report-dir | Direktorijum u koji će biti napisan izveštaj o doslednosti. |
| --check-graph | Provera doslednosti između čvorova, odnosa, svojstava, tipova i tokena. |
| --check-indexes | Provera doslednosti indeksa. |
| --check-index-structure | Provera strukture indeksa. |
| --additional-config | Konfiguracioni fajl za pružanje dodatnih ili zamenu postojećih konfiguracionih podešavanja u datoteci neo4j.conf. |
| --include-metadata | Metapodaci sadrže bezbednosne postavke vezane za bazu podataka. Ne može se koristiti za pravljenje rezervnih kopija sistemske baze podataka.   * roles - komande za kreiranje uloga i privilegija (i za bazu podataka i za graf) koje utiču na korišćenje baze podataka. * users - komande za kreiranje korisnika koji mogu da koriste bazu podataka i njihove uloge. * all - uključuju roles i users komande. |
| --prepare-restore | Oporavak rezervne kopije tako što će se primeniti najnovije povučene transakcije. Ako je onemogućeno, rezervna kopija će biti brža, ali će kasnije biti potreban oporavak rezervne kopije pre vraćanja podataka. |
| --parallel-recovery | Dozvola da više niti paralelno primenjuje transakcije na rezervnu kopiju. Za neke baze podataka i radna opterećenja, ovo može značajno smanjiti vreme izvršenja. |

*Tabela 1: neo4j-admin backup komanda*

U zavisnosti od toga da li je rezervna kopija bila uspešna ili ne, neo4j-admin backup komanda izlazi sa različitim kodovima. Kodovi grešaka uključuju detalje o tome do koje je greške došlo.

|  |  |
| --- | --- |
| 0 | Uspešno |
| 1 | Neuspešno |
| 2 | Pravljenje rezervne kopije je bilo uspešno, ali provera doslednosti nije prošla uspešno. |
| 3 | Pravljenje rezervne kopije je bilo uspešno, ali provera doslednosti je pronašla nedoslednosti. |

*Tabela 1: exit kodovi neo4j-admin backup komande*

Kada je u pitanju sama memorija koja se opredeljuje za pravljene rezervne kopije dostupne su opcije za konfigurisanje veličine gomile HEAP\_SIZE čime se konfiguriše maksimalna veličina gomile dodeljena procesu rezervne kopije i opcije za konfiguraciju keš stranice za rezervnu kopiju. Veličina keš stranice može se konfigurisati korišćenjem --pagecache sufiksa neo4j-admin backup komande. Ako nije eksplicitno definisano, veličina keš stranice je podrazumevano 8MB. Trebalo bi Neo4j kešu stranice dati što je moguće više memorije, sve dok je zadovoljeno ograničenje:

*Neo4j keš stranica + OS keš stranice < dostupna RAM memorija*

Pri čemu bi 2 do 4 GB trebalo da bude posvećeno kešu stranica operativnog sistema.

Na primer, ako trenutna baza podataka ima ukupnu mapiranu veličinu od 128 GB prema debug.log, i postoji dovoljno slobodnog prostora (što znači da po strani ima 2 do 4 GB za OS), onda se –pagecache može podesiti na 128GB.

Provera doslednosti rezervne kopije je glavna operacija koja može da potroši značajne računarske resurse prilikom pravljenja rezervne kopije, kao što su memorija, CPU, I/O. Kada se kreira rezervna kopija baze podataka onlajn, provera konzistentnosti se podrazumevano poziva na kraju procesa. Zbog toga je preporučljivo napraviti rezervnu kopiju i proveru konzistentnosti na namenskoj mašini, koja ima dovoljno slobodnih resursa, kako bi se izbegao negativan uticaj na rad servera. Alternativno, operacija pravljenja rezervne kopije može se odvojiti od provere doslednosti (koristeći neo4j-admin opciju rezervne kopije --check-consistency = false) i zakazati da se taj deo toka posla desi kasnije u vremenu, na namenskoj mašini. Provera doslednosti rezervne kopije je od vitalnog značaja za zaštitu i obezbeđivanje kvaliteta podataka i ne treba je potcenjivati.

Obezbeđivanje rezervne mrežne komunikacije pomoću SSL politike i zaštitnog zida štiti podatke od neželjenog upada i curenja. Kada se koristi komanda neo4j-admin backup, može se konfigurisati rezervni server tako da zahteva SSL/TLS, a klijent rezervne kopije da koristi kompatibilnu politiku.

U topologiji klastera, moguće je napraviti rezervnu kopiju sa bilo kog servera, a svaki server ima dva konfigurabilna porta koji mogu da služe rezervnoj kopiji. Ovi portovi su konfigurisani pomoću dbms.backup.listen.address i causal\_clustering.transaction\_listen\_address respektivno. Funkcionalno, oni su ekvivalentni za rezervne kopije, ali njihovo razdvajanje može omogućiti određenu operativnu fleksibilnost, dok korišćenje samo jednog porta može pojednostaviti konfiguraciju. Generalno se preporučuje izbor Read Replicas (kopija primarne instance koja odražava promene primarne u skoro realnom vremenu, u normalnim okolnostima) da deluju kao rezervni serveri. Štaviše, mogućnost problema sa performansama na Read Replici, uzrokovanih velikom rezervnom kopijom, neće uticati na performanse ili redundantnost. Ako replika za čitanje nije dostupna, onda se jezgro može izabrati na osnovu faktora, kao što su fizička blizina, propusni opseg, performanse i živost.

Primeri kako napraviti rezervnu kopiju jedne baze podataka i rezervnu kopiju više baza podataka korišćenjem komande neo4j-admin backup:

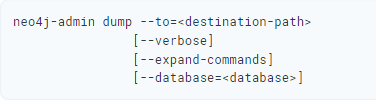
Jedna baza: *bin/neo4j-admin backup --backup-dir=/mnt/backups/neo4j --database=neo4j*

Više baza: *neo4j-admin backup --from=192.168.1.34 --backup-dir=/mnt/backups/neo4j --database=n\* --pagecache=4G*

Ovde samo napomena da ciljni direktorijum /mnt/backups/neo4j mora postojati pre pozivanja komande i baze podataka moraju biti onlajn.

## Rezervna kopija oflajn baze podataka

Rezervna kopija u oflajn režimu Neo4j baze podataka može se napraviti korišćenjem neo4j-admin dump komande. Ova komanda se može koristiti za izvođenje potpune rezervne kopije oflajn baze podataka. Baza podataka se prebacuje u arhivu sa jednom datotekom, nazvanom <database>.dump. Alternativno, neo4j-admin dump može da strimuje dump na standardni izlaz, omogućavajući da se izlaz prenese u drugi program, na primer u neo4j-admin load. Komanda se može pokrenuti samo lokalno sa onlajn ili oflajn Neo4j DBMS. Ne podržava SSL/TLS, za razliku od pravljenja onlajn rezvervne kopije. Sintaksa je slična kao i kod onlajn pravljenja rezervne kopije i prikazana je na slici 2:



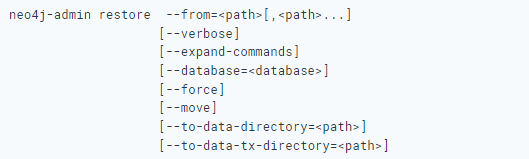
*Slika 2: Sintaksa komande neo4j-admin dump za pravljenje oflajn rezervne kopije*

U nastavku je primer kako napraviti dump default-ne baze podataka neo4j, pod nazivom neo4j-<timestamp>.dump, koristeći neo4j-admin dump komandu. Ciljni direktorijum /dumps/neo4j mora postojati pre pokretanja komande:

*bin/neo4j-admin dump --database=neo4j --to=/dumps/neo4j/neo4j-<timestamp>.dump*

## Oporavak baze podataka

Rezervna kopija baze podataka ili oflajn baza podataka se mogu povratiti pomoću komande restore neo4j-admin-a. Mora postojati kreirana baza podataka (CREATE DATABASE u odnosu na sistemsku bazu podataka) nakon što se operacija vraćanja završi, osim ako ne zamenjujete postojeću bazu podataka. neo4j-admin restore se mora pozvati kao neo4j user da bi se obezbedile odgovarajuće dozvole za fajl, što je naglašeno i ranije u radu. Sintaksa komande neo4j-admin restore je prikazana na slici 3:



*Slika 3: neo4j-admin restore komanda*

U tabeli ispod su objašnjena značenja sufiksa komande neo4j-admin restore:

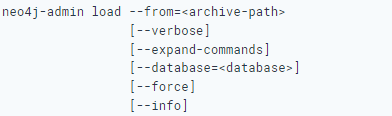
|  |  |
| --- | --- |
| --from | Putanja ili više putanja do rezervne kopije baze podataka za vraćanje. Putanja može da sadrži zvezdice ili upitnike u poslednjoj podputanji, ali ne sme da sadrži zareze. Zarezi se koriste za razdvajanje više putanja. |
| --verbose | Omogućiti detaljan izlaz. |
| --expand-commands | Dozvola za proširenje komande u proceni vrednosti konfiguracije. |
| --database | Naziv za obnovljenu bazu podataka. |
| --force | Zamena postojeće baze podataka. |
| --move | Premestiti datoteke rezervnih kopija na odredište, umesto da se kopiraju. |
| --to-data-directory | Osnovni direktorijum za baze podataka. Upotreba ove opcije je dozvoljena samo ako parametar --from ukazuje na jedan direktorijum. |
| --to-data-tx-directory | Osnovni direktorijum za evidenciju transakcija. Upotreba ove opcije je dozvoljena samo ako parametar --from ukazuje na jedan direktorijum. |

*Tabela 2: neo4j-admin restore komanda*

U nastavku je dat primer kako izvršiti oporavak baze podataka rezervnom kopijom koristeći komandu neo4j-admin restore:

*bin/neo4j-admin restore --from=/mnt/backups/neo4j --database=neo4j –force*

Komanda neo4j-admin load učitava bazu podataka iz arhive kreirane komandom neo4j-admin dump. Alternativno, neo4j-admin load može da prihvati dump sa standardnog ulaza, omogućavajući mu da prihvati unos sa neo4j-admin dump-a ili drugog izvora. Komanda se može pokrenuti sa onlajn ili oflajn Neo4j DBMS-a. Ako se menja postojeća baza podataka, mora se isključiti pre nego što se pokrene komanda. U suprotnom, mora se kreirati baza podataka (koristeći CREATE DATABASE u odnosu na sistemsku bazu podataka) nakon što se operacija učitavanja završi. Naravno i ovde važi da neo4j-admin load komanda mora biti pozvana kao neo4j user da bi se obezbedile odgovarajuće dozvole za fajl. Sintaksa komande data je na slici 4:



*Slika 4: sintaksa komande neo4j-admin load*

Jedini sufiks ove komande koji se razlikuje od komande neo4j-admin restore jeste --info i on omogućava štampanje informacije o metapodacima o arhivskoj datoteci, kao što su broj datoteka, broj bajtova i format datoteke za učitavanje. U nastavku je dat primer kako učitati postojeću bazu podataka koristeći komandu neo4j-admin load:

*bin/neo4j-admin load --from=/dumps/neo4j/neo4j-<timestamp>.dump --database=neo4j --force*

# Zaključak

Iz svega navedenog do sada zaključujemo da je za svaku rezervnu kopiju važno da svoje podatke skladišti odvojeno od proizvodnog sistema, gde ne postoje uobičajene zavisnosti, i po mogućstvu van lokacije (off-site). Kako se same rezervne kopije čuvaju dugo vremena, dugovečnost arhivskog skladištenja treba uzeti u obzir kao deo samog planiranja rezervnih kopija te na taj način obezbediti efikasno funkcionisanje sistema i bezbedan oporavak sistema ako dođe do nepredviđenih situacija. Isto tako, provera doslednosti rezervne kopije je od vitalnog značaja za zaštitu i obezbeđivanje kvaliteta podataka i ne treba je potcenjivati već je treba uvrstiti u jednu od krucijalnih stvari prilikom obezbeđivanja rezervne kopije baze podataka.

# Literatura

[1] Backup and restore

<https://neo4j.com/docs/operations-manual/current/backup-restore/>

[2] Neo4j data backup and restore

<https://blog.birost.com/a?ID=01800-7a672b7a-9012-436e-b742-c33babd31c14>

[3] NoSQL Undo: Recovering NoSQL Databases by Undoing Operations

<https://www.gsd.inesc-id.pt/~mpc/pubs/nosql-undo-recovering.pdf>

[4] THE REALITIES OF NOSQL BACKUPS

[https://trilio.io/wp-content/uploads/2015/04/Trilio-White-Paper\_The-Realities-of-NoSQL-Backups.pdf](https://trilio.io/wp-content/uploads/2015/04/Trilio-White-Paper_The-Realities-of-NoSQL-Backups.pdf%20)

[5] BARNS: Backup and Recovery for NoSQL Databases

[https://www.usenix.org/sites/default/files/conference/protectedfiles/hotstorage17\_slides\_kathpa.pdf](https://www.usenix.org/sites/default/files/conference/protectedfiles/hotstorage17_slides_kathpa.pdf%20)

[6] Tutorial: Back up and restore a single database in a single instance

<https://neo4j.com/docs/upgrade-migration-guide/current/tutorials/online-backup-restore-database/>

[7] Neo4j - Database Backup and Restore

<https://www.tutorialspoint.com/neo4j/neo4j_backup_restore.htm>

1. Neo4j Aura - Neo4j Aura je brza, skalabilna, always-on, potpuno automatizovana grafička platforma koja se nudi kao usluga u oblaku (cloud service). [↑](#footnote-ref-1)