

UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
KATEDRA ZA RAČUNARSTVO

BACKUP I RESTORE NEO4J BAZE PODATAKA

Predmet: Sistemi za upravljanje bazama podataka

Mentor:

doc. dr Aleksandar Stanimirović

Student:

Krstić Katarina, broj indeksa: 1400

SADRŽAJ

1. Uvod.....	3
2. Backup i restore	4
2.1. Tipovi backup-a	4
3. Backup i restore u Neo4j bazi podataka.....	7
3.1. Komande za backup i restore operacije u Neo4j-u	8
3.2. Neo4j backup režimi rada.....	8
3.3. Backup i restore online baze podataka.....	9
3.3.1. Backup online baze podataka	9
3.3.2. Konfiguracija	12
3.3.3. Primer backup-ovanja online Neo4j baze podataka	14
3.3.4. Priprema baze podataka za restore operaciju	15
3.3.5. Primer izvršenja pripreme Neo4j baze podataka za restore	16
3.3.6. Restore online baze podataka	16
3.3.7. Primer restore-ovanja Neo4j baze podataka.....	17
3.4. Backup i restore offline baze podataka	18
3.4.1. Backup offline baze podataka.....	18
3.4.2. Primer backup-ovanja offline baze podataka	19
3.4.3. Restore-ovanje offline baze podataka	19
3.4.4. Primer restore-ovanja offline Neo4j baze podataka	20
3.5. Kopiranje Neo4j baze podataka ili backup kopije Neo4j baze podataka.....	20
3.5.1. Primer kopiranja baze podataka.....	23
4. Zaključak.....	24
5. Literatura.....	25
6. Spisak slika.....	26

1. UVOD

Podaci predstavljaju osnovni, esencijalni deo projekata bilo koje veličine. Sa porastom samog sistema, najčešće raste i količina podataka kojima ovakvi sistemi manipulišu. Razvoj podataka i njihovo prikupljanje je proces koji traje.

Naročito poslednjih godina, informacije predstavljaju najvažnije sredstvo koje bi se moglo posedovati. Samim tim, podaci organizacije bilo koje vrste izloženi su ogromnom riziku. U isto vreme, načini na koje se može bilo kako eksterno, a neautorizovano, uticati na takve podatke su sve mnogobrojniji. Postoji puno mogućnosti za krađom, oštećenjem i malicioznim delovanjima nad podacima.

Pored toga, postoji i puno načina na koje bi moglo doći do nepovratnog gubitka važnih podataka.

Upravo na svemu ovome razvija se ideja o *backup*-u i *restore*-u. Naime, *backup* podataka predstavlja proces kojim bi se kreirala sigurna kopija podataka. Takva sigurna kopija skladištila bi se na bezbednom mestu. *Restore* je operacija suprotnog smera, odnosno, proces oporavka podataka.

Kompletna zamisao *backup* i *restore* operacija nudi, pre svega, sigurnost u to da, čak i da se nešto nepredviđeno desi, naši podaci neće biti izgubljeni nepovratno. U svakom momentu nam je dostupno da podatke oporavimo na osnovu sigurne kopije podataka kreirane na osnovu originalnih podataka.

Osim garancije da podaci neće biti izgubljeni, *backup* i *restore* nam omogućavaju i bezbednost podataka, jer takođe i svi sistemi koji se koriste u svrhe skladištenja ovakvih kopija garantuju sigurnost i čuvaju integritet podataka. Takođe, kako se ovim operacijama podaci prenose kroz mrežu, u većini slučajeva, koristi se *SSL* protokol koji garantuje enkripciju.

Sve ovo čini, pre svega, ideju, a onda i sam praktični aspekt *backup* i *restore* operacija veoma važnim i dragocanim za sisteme i projekte bilo koje veličine i vrste.

2. BACKUP I RESTORE

„A system failure wipes out all your data, corruption renders your data useless, an error leads to permanent deletion. Remember: Data loss always sneaks up on you.“

Među zahtevima najvišeg prioriteta bilo koje IT organizacije gotovo uvek se izdvaja zadatak koji se odnosi na zaštitu podataka, pre svega, sa kojim rame uz rame idu i zadaci koji se tiču očuvanja tih podataka od gubitka, krađe ili oštećenja bilo koje vrste i uzroka. Jedini način na koji se može osigurati očuvanje podataka, čak i u slučajevima kada bi se nešto od navedenog i desilo, jeste upravo *backup*-ovanje podataka.

Najjednostavnije rečeno, *backup* predstavlja redundantnu kopiju podataka sistema na koju se može osloniti u slučaju da nastane nerešiv problem u samom sistemu. Pre svega radi sigurnosti, ovaj postupak smatra se ključnim kada se razvija sistem bilo koje veličine i kompleksnosti. Pritom, princip je vrlo jednostavan: Napraviti kopiju podataka sistema ne bi li se u slučaju da postoji potreba, *restore*-ovale informacije koje su sačuvane.

Restore je postupak suprotan procesu *backup*-ovanja. Odnosno, *backup*-ovanjem mi čuvamo podatke na sigurnom mestu, dok ih *restore*-ovanjem vraćamo u naš sistem sa tog sigurnog mesta i na taj način oporavljamo oštećene ili izgubljene podatke.

2.1. TIPOVI BACKUP-A

Backup je naročito poslednjih godina postao sinonim za sigurnost podataka, a moguće je ostvariti ga na više načina. Takođe, postoji jako puno aplikacija koje nude mogućnost obavljanja backup i restore operacija na vrlo jednostavan način, korišćenjem jednostavnih naredbi ili još jednostavnijih grafičko-korisničkih interfejsa.

Osim odabira tipa *backup*-a koji će se vršiti, bitno je odabrati i lokaciju na kojoj će se backup kopija podataka čuvati. To se prevashodno odnosi na to da li će se podaci *backup*-ovati na *cloud*-u ili lokalno.

Postoji puno različitih podela kada je backup u pitanju, ali sve podele slažu se oko toga da postoje:

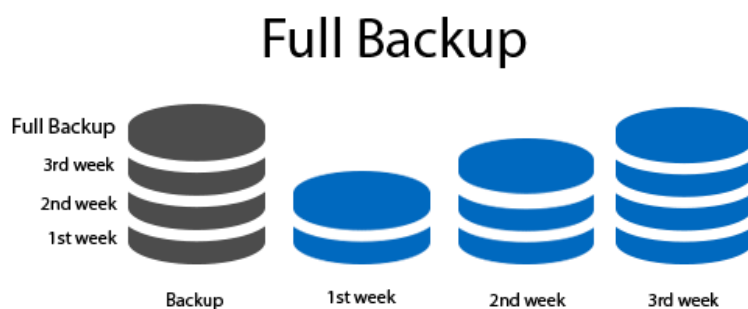
- *Full backup*
- *Incremental backup*
- *Differential backup*

Full backup

Full backup je najosnovniji i najjednostavniji tip *backup*-a. Kao što i samo ime govori, *full backup*-om se *backup*-uju apsolutno svi podaci. Takođe, *full backup* je i polazna tačka za sve druge tipove *backup*-a, odnosno, bez obzira na strategiju *backup*-ovanja koju bismo koristili, uvek je prvo neophodno odraditi jedan *full backup* da bismo mogli da iskoristimo bilo koji drugi.

Kao glavna prednost *full backup*-a izdvaja se ta da se restore u ovom slučaju najbrže obavlja. Međutim, sam proces *backup*-ovanja je duži i, pritom, potreban je veći skladišni prostor nego za preostale tipove *backup*-ovanja.

Uzevši sve stavke u obzir, zaključuje se da je *full backup* strategija koju bi bilo poželjno upotrebiti s vremena na vreme, a svakako kao polaznu tačku u ovom procesu.



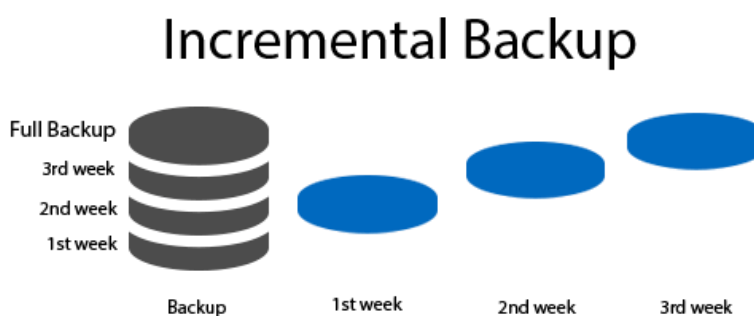
Slika1: Full backup

Incremental backup

Incremental backup strategija je, kao što je već pomenuto, strategija koja zahteva da pre nego se obavi, bude obavljen barem jedan *full backup*. Kao rezultat *incremental backup*-a kopiraju se isključivo oni podaci koji su izmenjeni u odnosu na poslednji izvršeni *backup*, bilo kog tipa.

Za razliku od *full backup* strategije, *incremental backup* ne zahteva ni toliko prostora ni toliko vremena za izvršenje, jer je količina podataka čiji se *backup* vrši *incremental backup*-om uvek manja. Ipak, kao glavni nedostatak, i ujedno razlika u odnosu na *full backup*, kod *incremental backup*-a je taj da je proces *restore*-ovanja u ovom slučaju najsporiji u odnosu na pomenute strategije. U ovom se slučaju zahteva *restore*-ovanje prvo najskorije izvršenog *full backup*-a, a zatim i svakog *incremental backup*-a izvršenog nakon njega i to istim redosledom kojim su se ti *incremental backup* i izvršavali.

Svakako, ova strategija se učestalije može izvršavati nego što bi se *full backup* strategija izvršavala.



Slika2: Incremental backup

Differential backup

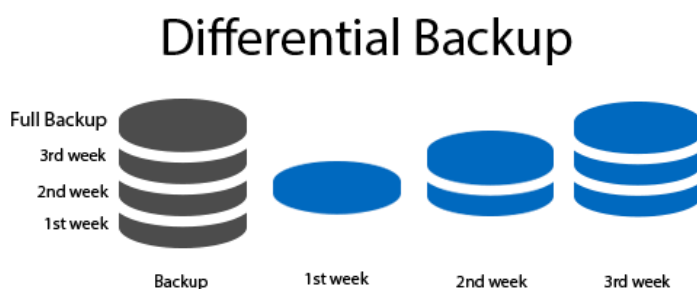
Kao što je moguće zaključiti *full backup* i *incremental backup* su sušte suprotnosti. Ono u čemu je jedna strategija dobra, drugoj predstavlja nedostatak i obrnuto. *Differential backup* strategija se tu nalazi kao kompromis između ove dve krajnosti. Teži da iskoristi prednosti i jedne i druge, a da pritom i nadomesti njihove nedostatke.

Naime, kao i *incremental backup*, i *differential backup* strategija zahteva da se prvo izvrši *full backup*, da bi mogao da se uopšte obavi.

U ovom se slučaju pravi kopija svih onih podataka koji su izmenjeni od poslednjeg *full backup*-a. Dakle, ne bilo kojeg, kao kod *incremental backup* strategije, već isključivo od poslednjeg *full backup*-a. To znači da bi prvim izvršenjem *differential backup*-ovanja bila napravljena kopija podataka izmenjenih tada, a zatim sledećim, između kojih se nije izvršio nijedan *full backup*, i kopija tih izmenjenih podataka i podataka izmenjenih nakon tog *differential backup*-a.

Ne bi li se uspešno *restore*-ovali podaci *backup*-ovani ovom strategijom, potrebno je obaviti *restore* najskorije izvršene *full backup* operacije i najskorije izvršene *differential backup* operacije.

Samim tim, možemo i da zaključimo da se *differential backup* strategijom vrši skladištenje veće količine podataka nego što je to slučaj kod *incremental backup*-a, ali se u isto vreme skladišti znatno manja količina podataka nego kod *full backup*-a. Dakle, potreban je veći skladišni prostor i više vremena za izvršenje *differential backup*-a od *incremental backup*-a, ali manje nego *full backup*-a. Sa druge strane, kada je *restore* proces u pitanju, *differential backup* strategijom potrebno je više vremena da bi se svi podaci *restore*-ovali nego što je potrebno kod *full backup*-a, ali umnogome manje nego što je potrebno kod *incremental backup*-a.



Slika3: Differential backup

3. BACKUP I RESTORE U NEO4J BAZI PODATAKA

Kada je reč o *backup*-ovanju baza podataka kakva je *Neo4j*, osim činjenice da se na taj način održava sigurnost podataka, omogućava se i jednostavnije premeštanje baze podataka, *upgrade*-ovanje iste ili proširenje neophodnog skladišnog prostora.

Pre obavljanja same *backup* operacije neophodno je definisati i *backup* strategiju na koju može uticati mnoštvo faktora, od kojih su samo neki količina podataka, dostupni sistemski resursi, broj baza podataka, okruženje(*development, test, production*), tolerancija na gubitak podataka, frekvencija *update*-ovanja baze podataka, zahtevi koji se tiču performansi tokom izvršenja *backup* i *restore* operacija, količina *backup* kopija koje bi trebalo skladištiti, lokacija na kojoj će se skladištiti, način na koji će se testirati operacija *restore* i koliko često, itd.

Tokom formiranja strategije i planiranja *backup* procesa neophodno je imati tačno i jasno definisan skup podataka čija bi *backup* kopija trebalo da se sačuva. Samim tim, kako *Neo4j* sistem za upravljanje bazama podataka omogućava skladištenje većeg broja baza podataka, svaku bi od njih bilo neophodno *backup*-ovati i to potpuno nezavisno od svih preostalih. Osim toga, ovaj *DBMS* sadrži i jednu sistemsku bazu podataka koja se koristi za skladištenje konfiguracionih informacija, kao što su operativna stanja baza podataka, konfiguracija koja se odnosi na sigurnost i bezbednost tih baza podataka, potencijalne definicije šema baza podataka, *login* informacije i role, odnosno uloge, korisnika. Kao i bilo koju drugu bazu podataka koja egzistira u ovakvom *DBMS*-u, tako bi i sistemsku bazu podataka trebalo *backup*-ovati i *restore*-ovati.

Osim samih baza podataka, bilo bi neophodno napraviti i sigurnu kopiju nekih važnih fajlova, kao što su *neo4j.conf* fajl pre svega, zatim sve informacije koje se odnose na enkripciju, ukoliko ona postoji(privatni ključ, digitalni sertifikat, i tako dalje), kao i bilo kakve vrste *plugin*-ova koji se koriste. A takođe, u slučaju da se koristi klaster, sve ove fajlove potrebno je *backup*-ovati za svaki od njih.

Na kraju, kao najočiglednije pitanje vezano za *backup* i *restore* proces neophodno je ustanoviti i koja će se lokacija(ili lokacije) koristiti za čuvanje kopija podataka. Pre svega, potrebno je obezbediti da se sam *backup* skladišni prostor nalazi na lokaciji koja je potpuno odvojena od lokacije *production* sistema, a najpreporučljivije je da se u ove svrhe koriste *cloud* sistemi i rešenja.

3.1. KOMANDE ZA BACKUP I RESTORE OPERACIJE U NEO4J-U

Neo4j dozvoljava izvršavanje *backup* operacije i nad *online* i nad *offline* bazom podataka. U te svrhe koristi *Neo4j Admin tool* komande, a korisnik koji ih izvršava mora ih izvršiti kao konkretni *Neo4j user*, ne bi li se na adekvatan način utvrdile permisije koje ima nad određenim fajlovima.

Komande koje bi se mogle koristiti za *backup* i *restore* operacije su:

- *neo4j-admin backup/restore* – ova naredba je podržana isključivo u *Enterprise* verziji, a koristi se za izvršavanje *backup* i *restore* operacija nad *online Neo4j* bazom podataka. Omogućava izvršavanje *backup*-a u *full* ili *incremental* režimu i pogodna je za *production* okruženja, s tim da zahteva veliku količinu memorije.
- *neo4j-admin dump/load* – ova naredba koristi se za izvršavanje *backup* i *restore* operacija nad *offline* bazom podataka, što ne umanjuje mogućnost da se koristi i za *online* bazu podataka. Pogodno ju je koristiti u okruženjima u kojima *downtime* ne bi predstavljao bitan faktor. Pritom, brža je od *backup* komande, a kako za rezultat kreira *archive dump* fajl, zahteva mnogo manje memorijskog prostora nego što je to slučaj kod *backup* komande.
- *neo4j-admin copy* – ova naredba koristi se za kopiranje i baze podataka ali i nekog drugog *backup*-a baze podataka. Pogodna je u slučajevima kada bi trebalo očistiti bazu podataka od nekonzistentnosti unutar iste, kada je neophodno zahtevati dodatni skladišni prostor, ili u slučaju da je neophodno *upgrade*-ovanje sa *3.x* na *4.x* verziju *Neo4j*-a.

3.2. NEO4J BACKUP REŽIMI RADA

Neo4j klijent može raditi u dva moguća *backup* režima, a to su *full* i *incremental backup* režimi.

Bez obzira na to koji će se od ova dva režima koristiti u kasnijem radu, *full backup* je režim koji prvi mora da bude izvršen i kada se radi o *online* i kada se radi o *offline* bazi podataka.

```
$neo4j-home> export HEAP_SIZE=2G
$neo4j-home> mkdir /mnt/backups
$neo4j-home> bin/neo4j-admin backup --from=192.168.1.34 --backup-dir=/mnt/backups/neo4j --database=neo4j --pagecache=4G
Doing full backup...
2017-02-01 14:09:09.510+0000 INFO [o.n.c.s.StoreCopyClient] Copying neostore.nodestore.db.labels
2017-02-01 14:09:09.537+0000 INFO [o.n.c.s.StoreCopyClient] Copied neostore.nodestore.db.labels 8.00 kB
2017-02-01 14:09:09.538+0000 INFO [o.n.c.s.StoreCopyClient] Copying neostore.nodestore.db
2017-02-01 14:09:09.540+0000 INFO [o.n.c.s.StoreCopyClient] Copied neostore.nodestore.db 16.00 kB
...
...
...
```

Slika4: Full backup nad *Neo4j* bazom podataka

Nakon izvršenog *full backup*-a na određenu lokaciju, svaki naredni pokušaj izvršenja *backup* operacije podrazumevano bi težio ka korišćenju *incremental* režima. To znači da bi težio ka tome da se izvrši *backup*-ovanje samo onih podataka čiji je sadržaj promenjen od poslednje izvršene *backup* operacije, bilo da je ona nastala *full* ili *incremental backup* operacijom.

Međutim, neće se u svakom slučaju izvršiti *incremental backup* u slučaju da je prvo izvršen *full backup*. Naime, prilikom poziva *backup* operacije na izvršavanje, vrši se provera *transaction log* fajlova, a na osnovu njih se zaključuje koje bi informacije trebalo *backup*-ovati *incremental backup* operacijom i to da li bi trebalo *backup* izvršiti u *incremental* ili *full backup* režimu. Ukoliko ovih fajlova nema ili iz nekog razloga nisu dostupni na serveru, sistem će podrazumevano ponovo krenuti sa izvršavanjem *backup* operacije u *full* režimu, odnosno *backup*-ovaće sve podatke.

Za razliku od *backup* operacije u *full* režimu, *backup* operaciju u *incremental* režimu moguće je izvršiti samo nad *online* bazom podataka.

```
$neo4j-home> export HEAP_SIZE=2G
$neo4j-home> bin/neo4j-admin backup --from=192.168.1.34 --backup-dir=/mnt/backups/neo4j --database=neo4j --pagecache=4G
Destination is not empty, doing incremental backup...
Backup complete.
```

Slika5: Incremental backup nad Neo4j bazom podataka

3.3. BACKUP I RESTORE ONLINE BAZE PODATAKA

3.3.1. BACKUP ONLINE BAZE PODATAKA

Za *backup online* baze podataka koristi se komanda:

neo4j-admin backup

i pritom je mora izvršiti *neo4j* korisnik ne bi li se adekvatno ustanovile permisije nad fajlovima sistema. Ova komanda može se koristiti i za *backup* operaciju u *full*, ali i u *incremental* režimu.

Jedna od najvažnijih karakteristika ove komande jeste ta da ona podrazumevano nakon izvršenja *backup* operacije izvršava i proveru konzistentnosti same baze podataka i njene sigurne, *backup*-ovane, kopije dobijene kao rezultat izvršenja same naredbe.

Sa druge strane, kao njen glavni nedostatak izdvaja se velika potrošnja sistemskih resursa, pre svega memorije i procesora, pa se iz tog razloga, za izvršavanje *backup* operacije ovom naredbom, najčešće koristi potpuno nezavisna mašina, namenjena upravo za to.

```
neo4j-admin backup --backup-dir=<path>
                    [--verbose]
                    [--expand-commands]
                    [--from=<host:port>]
                    [--database=<database>]
                    [--fallback-to-full=<true/false>]
                    [--pagecache=<size>]
                    [--check-consistency=<true/false>]
                    [--report-dir=<path>]
                    [--check-graph=<true/false>]
                    [--check-indexes=<true/false>]
                    [--check-index-structure=<true/false>]
                    [--check-label-scan-store=<true/false>]
                    [--check-property-owners=<true/false>]
                    [--additional-config=<path>]
                    [--include-metadata=<all/users/roles>]
                    [--prepare-restore=<true/false>]
                    [--parallel-recovery=<true/false>]
```

Slika6: Sintaksa neo4j-admin backup komande

Naredbu za *backup online* baze podataka moguće je definisati mnoštvom parametara:

- **--backup-dir=<path>** - opcija kojom se specificira lokacija *backup* kopije. Ovaj parametar jedini nije opcioni i obavezno ga je definisati prilikom poziva komande.
- **--verbose** – opcija kojom se omogućava prikaz detaljnijih informacija tokom izvršavanja naredbe. Ovaj parametar je opcioni.
- **--expand-commands** – opcija kojom se omogućava *extract*-ovanje komande tokom evaluacije *config* vrednosti. Ovaj parametar je opcioni.
- **--from=<host:port>** - opcija kojom se specificiraju *host* i *port* Neo4j-a. Podrazumevana vrednost ovog parametra je *localhost:6362* i opcioni je.
- **--database=<database>** - opcija kojom se specificira naziv baze podataka koju bi trebalo *backup*-ovati ovom komandom, odnosno nazivi baza podataka, s obzirom na to da je moguće definisati veći broj vrednosti za ovaj parametar. Prilikom definisanja vrednosti ovog parametra moguće je koristiti specijalne karaktere ? i * za formiranje paterna koji bi trebalo da zadovolje nazivi baza podataka koje bi trebalo *backup*-ovati. Takođe, ovaj parametar može imati i samo vrednost *, što bi značilo da bi trebalo izvršiti *backup* operaciju nad svim bazama podataka. Podrazumevana vrednost parametra *database* je *neo4j* i on je opcioni.
- **--fallback-to-full=<true/false>** - opcija kojom se specificira način ponašanja same komande u slučaju da dođe do greške prilikom izvršavanja *backup* operacije u *incremental* režimu. Naime, u slučaju da ova komanda ne uspe, *backup* će biti premešten u <name>.err.<N> i zatim će preći na izvršenje *full backup* operacije. Naravno, ovo se dešava u slučaju da je ova opcija postavljena na *true* vrednost, što je ujedno i njena podrazumevana vrednost, dok, u slučaju da je njena vrednost *false*, *full backup* se neće izvršiti po neuspešnom izvršenju *incremental backup*-a.
- **--pagecache=<size>** - opcija kojom se specificira veličina *page* keša koji će biti korišćen tokom izvršavanja *backup* operacije. Podrazumevana vrednost ovog

parametra iznosi *8MB*, što znači da, ukoliko se on ne specificira sa nekom drugom vrednošću, koristiće se *8MB* u ove svrhe.

- **--check-consistency=<true/false>** - opcija kojom se specificira ponašanje komande nakon što se okonča *backup* proces u kontekstu provere konzistentnosti podataka između originalne baze podataka i njene kopije kreirane ovom komandom. Ukoliko je vrednost ovog parametra postavljena na *true*, nakon izvršene *backup* operacije će se ispitati i proveriti konzistentnost podataka, što će se desiti i podrazumevano, s obzirom na to da je *true* podrazumevana vrednost ovog parametra. U slučaju da se vrednost ove opcije postavi na *false*, ova provera se neće vršiti.
- **--report-dir=<path>** - opcija kojom se specificira lokacija na kojoj će biti ispisan i na kojoj će se skladištiti izveštaj izvršenja sprovedene *backup* operacije. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--check-graph=<true/false>** - opcija kojom se specificira da li će biti obavljena provera konzistentnost čvorova, relacija, atributa, tokena i tipova podataka po okončanju izvršenja *backup* operacije. Naravno, u slučaju da je vrednost ovog parametra postavljena na *true*, što je i podrazumevana vrednost ove opcije, ova provera biće obavljena, dok u suprotnom to neće biti slučaj.
- **--check-index-structure=<true/false>** - opcija kojom se specificira da li će biti obavljene provere po pitanju strukture indeksa. U slučaju da je vrednost ovog parametra postavljena na *true*, što je i podrazumevana vrednost ove opcije, provera će se obaviti, dok u suprotnom to neće biti slučaj.
- **--check-label-scan-store=<true/false>** - ova opcija je *deprecated* i ignoriše se.
- **--check-property-owners=<true/false>** - ova opcija je *deprecated* i ignoriše se.
- **--additional-config=<path>** - opcija kojom se specificira dodatni fajl koji će se koristiti da definiše dodatne ili prepíše postojeće konfiguracione informacije u *neo4j.conf* fajlu. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--include-metadata=<all/users/roles>** - opcija kojom se specificiraju metapodaci koje bi trebalo uključiti u proces *backup*-ovanja. Metapodaci na koje se odnosi ovaj parametar sadrže važne sigurnosne informacije u vezi sa bazom podataka. Ova opcija može se definisati za bilo koju bazu podataka koju je neophodno *backup*-ovati, osim za sistemsku bazu podataka. Moguće vrednosti ovog parametra su *all*, *users* i *roles*, pri čemu *roles* označava komande za kreiranje uloga i privilegija koje utiču na upotrebu baze podataka; *users* označava komande za kreiranje korisnika koji bi mogli koristiti bazu podataka i njima dodeljene uloge; dok *all* uključuje i jedne i druge.
- **--prepare-restore=<true/false>** - opcija kojom se specificira da li bi trebalo izvršiti pripremu baze podataka za *restore* operaciju primenom poslednjih *pull*-ovanih transakcija. U slučaju da je vrednost ovog parametra postavljena na *false*, sam *backup* proces bio bi brži, dok bi proces *restore* operacije bio značajno sporiji, zato što se ovaj oporavak zahteva pre samog *restore*-a. Takođe, vrlo karakteristično je to da ukoliko je ovaj parametar postavljen na *false* vrednost, i sam *check-consistency* parametar će implicitno dobiti vrednost *false*, jer nije moguće izvršiti proveru konzistentnosti originalne baze podataka i ovako dobijene njene kopije, jer podaci u njoj nisu u formatu koji je adekvatan za oporavak originalne baze podataka. Podrazumevana vrednost ovog parametra je *true* i on je opcioni.

- **--parallel-recovery=<true/false>** - opcija kojom se dozvoljava da veći broj niti istovremeno obavlja *backup* operaciju. Ovo je eksperimentalna opcija za čiji rad bi se pre upotrebe trebalo konsultovati sa *Neo4j* ekspertima, ne bi li došlo do greške tokom rada. Podrazumevana vrednost ovog parametra je *false* i on je opcioni.

Rezultat ove komande je sistemski kod. Kada je reč o kodovima koji se odnose na *error*, uz sam kod uvek se navode i dodatne informacije koje pružaju uvid u to zbog čega je do same greške došlo. Takođe, u zavisnosti od toga da li se *backup* operacija izvršava nad jednom ili nad većim brojem baza podataka, možemo imati različite sistemske kodove kao rezultat.

Neo4j backup exit kodovi u slučaju da se *backup* operacija izvršava nad jednom bazom podataka mogu biti:

- **0 – Success** – kod koji označava da je *backup* operacija okončana uspešno
- **1 – Backup failed** – kod koji označava da *backup* operacija nije uspešno okončana
- **2 – Backup succeeded but consistency check failed** – kod koji označava da je *backup* operacija uspešno završena, ali da provera konzistentnosti nije
- **3 – Backup succeeded but consistency check found inconsistencies** – kod koji označava da je *backup* operacija uspešno završena, ali da je proverom konzistentnosti utvrđeno da postoje određene nekonzistentnosti između originalne baze podataka i njene kopije kreirane ovom komandom.

Neo4j backup exit kodovi u slučaju da se *backup* operacija izvršava nad većim brojem baza podataka mogu biti:

- **0 – All databases are backed up successfully** – kod koji označava da je uspešno okončana *backup* operacija svih baza podataka koje je trebalo *backup*-ovati.
- **1 – One or several backups failed** – kod koji označava da *backup* operacija nije uspeła u jednoj ili većem broju baza podataka koje je trebalo *backup*-ovati.

3.3.2. KONFIGURACIJA

Konfiguracija servera

Parameter name	Default value	Description
<code>dbms.backup.enabled</code>	<code>true</code>	Enable support for running online backups.
<code>dbms.backup.listen_address</code>	<code>127.0.0.1:6362</code>	Listening server for online backups.

Slika7: Konfiguracija servera

Parametri na slici predstavljaju osnovne relevantne parametre za konfiguraciju servera u slučaju izvršavanja *backup* operacije. Podrazumevano, backup server je *enable*-ovan, s tim da podrazumevano sluša na *127.0.0.1*, što bi bilo neophodno izmeniti u slučaju da se koristi druga mašina.

Konfiguracija memorije

Kada je reč o konfiguraciji memorije za proces *backup* operacije, moguće je manipulirati parametrima: *heap size* i *cache size*.

- **Heap size** – ovim parametrom definiše se maksimalna veličina *heap*-a alocirana za obavljanje *backup* operacije. Postavlja se definisanjem vrednosti *environment* varijable *HEAP_SIZE* pre startovanja same operacije, dok, u slučaju da nema postavljenu vrednost, *Java Virtual Machine* tu vrednost određuje sama na osnovu sistemskih resursa.
- **Page cache** – ovaj parametar definiše se u okviru samog poziva komande njenim opcionim parametrom. Ukoliko se ne definiše nijedna druga njegova vrednost, koristiće se podrazumevana vrednost ovog parametra koja iznosi *8MB*.

Napisano je pravilo da bi za *page cache* trebalo postaviti maksimalnu moguću vrednost koja bi zadovoljila uslov:

$$\text{Neo4j page cache} + \text{OS page cache} < \text{available RAM}$$

Konfiguracija sistemskih resursa

Provera konzistentnosti podataka veoma je značajan zadatak koji iziskuje veliku količinu najznačajnijih sistemskih resursa: memorije, procesora i *I/O*. Ova provera podrazumevano je uključena u rad same komande prilikom njenog poziva, a to se može i promeniti definisanjem vrednosti *--check-consistency* opcije na *false*. Međutim, kako provera konzistentnosti predstavlja veoma važan korak u kompletnoj ideji *backup* i *restore* procesa, svakako bi je trebalo obaviti ako ne pri samom izvršenju *backup*-ovanja, onda u nekom bliskom trenutku.

Važnost konzistentnosti podataka i očuvanje i provera iste ne bi trebalo da se zanemari, jer se na taj način podaci čuvaju na adekvatan način i pritom se na taj način vodi računa i o samom kvalitetu podataka koji se čuvaju.

Upravo sa gledišta sistemskih resursa, neretko je praksa da se *backup* operacija izvršava na posebnoj, nezavisnoj mašini, koja je namenjena upravo za to. Na taj način, mašina koja bi bila korišćena u svrhe razvoja sistema imala bi mogućnost da koristi svoje sistemske resurse u punoj snazi bez problema, iako bi se u tom slučaju vršila *backup* operacija. Ovim pristupom omogućava se da u svakom *backup* procesu bude izvršena i provera konzistentnosti, jer ne bi postojala potreba za štednjom sistemskih resursa.

Konfiguracija bezbednosti

Kada govorimo o *backup* i *restore* operacijama, suštinski se ideja na kojoj se ovi procesi zasnivaju ne odnosi isključivo na situacije u kojima bi trebalo očuvati samo kvalitet podataka, odnosno same tačne informacije, od gubitka, već ih treba sačuvati i od bilo kakve zloupotrebe ili krađe. Samim tim, neophodno je voditi računa o bezbednosti podataka i tokom

samih procesa izvršenja *backup* i *restore* operacija, ali i o tome da lokacije na kojima se podaci čuvaju garantuju bezbednost i sigurnost.

Neo4j nudi mogućnost upotrebe *SSL framework*-a i na taj način osigurava da mrežna komunikacija koristi *SSL* protokol. Na taj način, podaci se šire tokom izvršenja *backup* i *restore* operacija.

Topology	Backup target address on database server	SSL policy setting on database server	SSL policy setting on backup client	Default port
Standalone instance	<code>dbms.backup.listen_address</code>	<code>dbms.ssl.policy.backup</code>	<code>dbms.ssl.policy.backup</code>	6362
Causal cluster	<code>dbms.ssl.policy.cluster_causal_clustering.transaction_listen_address</code>	<code>dbms.ssl.policy.cluster</code>	<code>dbms.ssl.policy.backup</code>	6000

Slika8: Konfiguracija bezbednosti

Veoma je važno da ne postoji eksterni pristup portu koji se definiše kao vrednost `dbms.backup.listen_address`-a, jer bi to značilo da bi sami podaci bili dostupni neautorizovanom korisniku da kopira kompletan *backup* sadržaj. Iz tog razloga, najbolje bi bilo da bilo kakav eksterni pristup backup portu bude blokiran od strane *firewall*-a.

Konfiguracija klastera

Kada se koristi topologija klastera, moguće je preuzeti *backup* sa bilo kojeg servera. Pritom, svaki od tih servera definiše dva porta koji se mogu koristiti da opsluže sam *backup*. Ti portovi definišu se pomoću `dbms.backup.listen_address` i `casual_clustering.transaction_listen_address`. Oni su za samu *backup* operaciju potpuno ekvivalentni.

Korišćenje i definisanje samo jednog od dva porta olakšava konfiguraciju, dok korišćenje oba ova porta nudi, pre svega, operativnu funkcionalnost.

Takođe, kada je reč o klasterima, dobra je praksa selektovati *Read Replica* instance da budu korišćene za *backup*, ukoliko su dostupne. Na ovaj način bi se čak i prilikom zahtevnih *backup* operacija, koje bi zahtevale puno resursa i vremena, *Core* sistem maksimalno poštdeo sa stanovišta performansi.

3.3.3. PRIMER BACKUP-OVANJA ONLINE NEO4J BAZE PODATAKA

Na slici *Slika9* predstavljena je komanda kojom bi se omogućilo *backup*-ovanje jedne baze podataka, i to baze podataka čiji je naziv *neo4j*. Iza same komande ***neo4j-admin backup*** nalazi se njen jedini neopcionni parametar kojim se definiše lokacija na kojoj će se skladištiti *backup* kopija.

Preduslov da se ova komanda obavi na način na koji je definisana na slici jeste da lokacija na kojoj će se skladištiti ova *backup* kopija postoji pre nego se komanda pozove na izvršenje.

```
bin/neo4j-admin backup --backup-dir=/mnt/backups/neo4j --database=neo4j
```

Slika9: Primer poziva za backup-ovanje online Neo4j baze podataka neo4j-admin backup komandom

Na slici Slika10 predstavljena je komanda kojom bi se izvršilo *backup*-ovanje većeg broja baza podataka. Komandom definisanom na ovaj način bila bi *backup*-ovana svaka baza podataka čiji naziv počinje slovom *n*. U slučaju ove naredbe, suštinski je važno to da određena lokacija postoji pre nego što se ona pozove na izvršenje i, naravno, da sve ove baze podataka budu *online*.

```
neo4j-admin backup --from=192.168.1.34 --backup-dir=/mnt/backups/neo4j --database=n* --pagecache=4G
```

Slika10: Primer poziva za backup-ovanje većeg broja online Neo4j baza podataka neo4j-admin backup komandom

3.3.4. PRIPREMA BAZE PODATAKA ZA RESTORE OPERACIJU

Može se desiti da se *backup* operacija izvršava tako da joj je *--prepare-restore* opcija setovana na vrednost *false*. To znači da se u samoj konačnoj *backup* kopiji informacije ne nalaze u onoj formi u kojoj bi se mogle iskoristiti za *restore* operaciju i da se ne može ustanoviti konzistentnost sa originalnom bazom podataka.

Ukoliko je to slučaj, pre nego se izvrši *restore* operacija, potrebno je pripremiti *backup* kopiju za to. U te svrhe koristi se posebna komanda:

neo4j-admin prepare-restore

Zadatak ove komande je da primeni sve transakcije i time pripremi sadržaj skladišnog prostora za *restore* operaciju.

```
neo4j-admin prepare-restore --target=<path>[,<path>...]...
                             [--verbose]
                             [--expand-commands]
                             [--parallel-recovery]
```

Slika11: Sintaksa neo4j-admin prepare-restore komande

- ***--target*** – ovo je jedini parametar *prepare-restore* komande koji nije opcioni, a koristi se da definiše putanju do same *backup* kopije koja se priprema za *restore* operaciju. Prilikom definisanja ove putanje, dozvoljeno je korišćenje specijalnih karaktera *?* i *** radi definisanja paterni koji bi putanja(ili putanje) trebalo da zadovolje. Dozvoljeno je pritom i korišćenje podstringova koji bi olakšali definisanje takvog paterni, ali u njima ne sme da se nalazi karakter zapete. To je zbog toga što se različite putanje koje mogu biti vrednosti ovog parametra odvajaju upravo tim specijalnim karakterom, pa bi to dovelo i do pogrešnog tumačenja.
- ***--verbose*** – ovom opcijom omogućio bi se detaljni prikaz izvršavanja same komande. Ovaj parametar je opcioni.

- **--expand-commands** – opcija kojom se omogućava *extract*-ovanje komande tokom evaluacije *config* vrednosti. Ovaj parametar je opcioni.
- **--parallel-recovery** – ova opcija omogućava da veći broj niti paralelno izvršava ovu operaciju. Podrazumevana vrednost parametra je *false* i on je opcioni.

3.3.5. PRIMER IZVRŠENJA PRIPREME NEO4J BAZE PODATAKA ZA RESTORE

Na slici je prikazana komanda kojom bi se obavila priprema sadržaja *backup* kopije *Neo4j* baze podataka na definisanoj putanji za *restore* operaciju. Iza same komande u ovom slučaju nije navedena nijedna od mogućih opcija, već je naveden jedino osnovni i glavni, neizostavni parametar same komande, kojim se definiše putanja na kojoj se ovakva *backup* kopija nalazi.

```
bin/neo4j-admin prepare-restore --target=/mnt/backups/neo4j
```

Slika12: Primer poziva *neo4j-admin prepare-restore* komande

3.3.6. RESTORE ONLINE BAZE PODATAKA

Komanda

neo4j-admin restore

koristi se za *restore*-ovanje *backup* kopije baze podataka, a može se primeniti i za *online* i za *offline* bazu podataka. Kao što je slučaj i sa *backup* operacijom, i *restore* operaciju je neophodno izvršiti kao *neo4j* korisnik, ne bi li se permisije nad fajlovima u sistemu utvrdile na adekvatan način.

Osim u slučaju kada se menja sadržaj postojeće baze podataka, potrebno je kreirati *restore*-ovanu bazu podataka nakon što se okonča izvršenje ove komande. To bi bilo moguće učiniti pozivom *CREATE DATABASE* naredbe nad sistemskom bazom podataka.

Kao što je pomenuto, da bi *restore* operacija uopšte mogla da se izvrši, neophodno je da sama kopija bude pripremljena za to, što će svakako biti slučaj ukoliko su se *backup* operacije izvršavale sa aktivnom *--prepare-restore* opcijom. U suprotnom, zahteva se izvršenje komande za pripremu *backup* kopije za *restore* operaciju.

```
neo4j-admin restore --from=<path>[,<path>...]
                    [--verbose]
                    [--expand-commands]
                    [--database=<database>]
                    [--force]
                    [--move]
                    [--to-data-directory=<path>]
                    [--to-data-tx-directory=<path>]
```

Slika13: Sintaksa *neo4j-admin restore* komande

- **--from=<path>** - ovaj parametar komande *restore* jedini nije opcioni, a koristi se da definiše putanju do *backup* kopije ili *backup* kopija baza podataka čiji bi *restore* trebalo izvršiti. Moguće je, dakle, definisati veći broj putanja koje bi bile vrednosti ovog parametra. U samoj definiciji putanje, odnosno vrednosti ovog parametra, dozvoljeno je korišćenje specijalnih karaktera ? i *, kao i podstringova, ne bi li se sam patern koji bi putanja trebalo da zadovolji, lakše specificirao. Pritom, podstringovi ne bi smeli da sadrže specijalni karakter zapete, jer se upravo ovaj specijalni karakter koristi za razdvajanje različitih vrednosti ovog parametra, pa bi to moglo dovesti do pogrešnog tumačenja vrednosti parametra.
- **--verbose** – ovaj parametar koristi se da omogući pružanje detaljnih informacija o izvršenju same komande. Parametar je opcioni.
- **--expand-commands** – opcija kojom se specificira da li će se izvršiti *extract*-ovanje komande tokom evaluacije *config* vrednosti. Parametar je opcioni.
- **--database=<database>** - ovaj parametar komande specificira naziv *restore*-ovane baze podataka. Parametar je opcioni i njegova podrazumevana vrednost je *neo4j*.
- **--force** – ovaj parametar specificira to da li će rezultat izvršavanja ove komande dovesti do zamene postojeće baze podataka ili ne. Parametar je opcioni.
- **--move** – ovaj parametar specificira to da li će se *backup*-ovani fajlovi premestiti ili kopirati do određene lokacije. Navođenje ove opcije dovešće do premeštanja fajlova, a sam parametar je opcioni.
- **--to-data-directory=<path>** - osnovni, ili bazni direktorijum baza podataka. Ovaj parametar može se postaviti isključivo u slučaju da *from* parametar pokazuje na jedan folder. Parametar je pritom opcioni i nema podrazumevanu vrednost.
- **--to-data-tx-directory=<path>** - osnovni, ili bazni direktorijum u kojem se nalaze *transaction logs* fajlovi. Takođe, kao i prethodni, i ovaj parametar moguće je postaviti isključivo u slučaju da *from* parametar pokazuje na jedan folder. Pritom, parametar je opcioni i nema podrazumevanu vrednost.

3.3.7. PRIMER RESTORE-OVANJA NEO4J BAZE PODATAKA

Na slici *Slika14* nalazi se komanda kojom bi se izvršila *restore* operacija nad *backup* kopijom na definisanoj putanji. Osim u slučaju da se ne vrši zamena postojeće baze podataka, pošto se okonča izvršenje ove komande, potrebno je izvršiti i *CREATE DATABASE* naredbu nad sistemskom bazom podataka.

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

Slika14: Primer restore-ovanja Neo4j baze podataka neo4j-admin restore komandom

Prilikom *backup*-ovanja baza podataka mogu se *backup*-ovati i određeni metapodaci, koji se odnose na korisnike i njihove uloge, odnosno permisije unutar sistema. Ovi podaci bi se manualno, nezavisno od same *neo4j-admin restore* operacije morali *restore*-ovati. Ne bi li se to obezbedilo, neophodno je pristupiti *<neo4j-home>* direktorijumu i u njemu izvršiti skriptu koja nastaje kao rezultat izvršenja *neo4j-admin restore* operacije.

data/scripts/databasename/restore_metadata.cypher

```
cat data/scripts/databasename/restore_metadata.cypher | bin/cypher-shell -u user -p password -a ip_address:port -d system --param "database => 'databasename'"
```

Slika15: Primer izvršenja skripte za kreiranje baze podataka nakon restore operacije u Linux-u

```
type data\scripts\databasename\restore_metadata.cypher | bin\cypher-shell.bat -u user -p password -a ip_address:port -d system --param "database => 'databasename'"
```

Slika16: Primer izvršenja skripte za kreiranje baze podataka nakon restore operacije u Windows-u

3.4. BACKUP I RESTORE OFFLINE BAZE PODATAKA

3.4.1. BACKUP OFFLINE BAZE PODATAKA

Neo4j baza podataka može biti *backup*-ovana u *offline mode*-u korišćenjem komande:

neo4j-admin dump

Kao rezultat ove naredbe dobija se jedan *archive* fajl *<database>.dump* ili se taj rezultat može proslediti na standardni izlaz mašine i time omogućiti da se na njega poveže drugi program koji bi ga čitao i koristio nakon toga.

Naime, ova komanda može se koristiti za *backup* operaciju u *full* režimu, nad *offline* bazom podataka, ali se može koristiti i za *backup*-ovanje *online* baze podataka, takođe. Za razliku od *backup* i *restore neo4j-admin* operacija, *dump* operacija ne pruža bezbednosnu zaštitu korišćenjem *SSL* protokola.

```
neo4j-admin dump --to=<destination-path>
                  [--verbose]
                  [--expand-commands]
                  [--database=<database>]
```

Slika17: Sintaksa neo4j-admin dump komande

- ***--to=<destination-path>*** - ovaj parametar je jedini neopcionni kada je ova komanda u pitanju i njime se definiše odredišna putanja, što može biti ili fajl ili folder, gde će se nalaziti *dump* baze podataka. U slučaju da se za vrednost ovog parametra postavi – to znači da će se rezultat izvršenja ove komande slati na standardni izlaz.
- ***--verbose*** – ovaj parametar je opcionni, a koristi se da pruži detaljne informacije o izvršenju same naredbe.
- ***--expand-commands*** – opcija kojom se specificira da li će se izvršiti *extract*-ovanje komande tokom evaluacije *config* vrednosti. Ovaj parametar je opcionni i nema podrazumevanu vrednost.
- ***--database=<database>*** - ovaj parametar predstavlja naziv baze podataka čiji će se *dump* izvršiti. Podrazumevana vrednost ovog parametra je *neo4j*, a kako je on opcionni, ukoliko se ne specificira i ne postavi mu se drugačija vrednost, koristiće se vrednost *neo4j*.

3.4.2. PRIMER BACKUP-OVANJA OFFLINE BAZE PODATAKA

Na slici se nalazi komanda kojom bi se izvršilo kreiranje *dump*-a *Neo4j* baze podataka. Preduslov za izvršenje komande u ovom formatu u kojem je navedena na slici jeste da destinacioni folder postoji pre nego se izvrši ova komanda, ne bi li se ona uspešno izvršila.

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

Slika18: Primer poziva za backup-ovanje *Neo4j* baze podataka *neo4j-admin dump* komandom

3.4.3. RESTORE-OVANJE OFFLINE BAZE PODATAKA

Komandom

neo4j-admin load

može se učitati baza podataka iz definisanog i kreiranog arhiviranog *dump* fajla, dobijenog izvršenjem *neo4j-admin dump* komandom, a može takođe učitati bazu podataka i iz sadržaja koji bi *dump* komanda poslala na standardni izlaz. Kao i *dump* komanda, i komanda *load* može se izvršiti i nad *online* i nad *offline* bazom podataka, a kao i sve preostale navedene naredbe, mora je izvršiti *neo4j* korisnik, ne bi li sve permisije adekvatno bile ustanovljene.

U slučaju da se postojeća baza podataka menja ovom komandom, potrebno ju je ugasiti pre nego se izvrši *load* naredba. U suprotnom, ukoliko takva baza podataka ne postoji, po izvršenju same komande, neophodno je izvršiti *CREATE DATABASE* naredbu nad sistemskom bazom podataka i time kreirati samu bazu podataka.

```
neo4j-admin load --from=<archive-path>
                  [--verbose]
                  [--expand-commands]
                  [--database=<database>]
                  [--force]
                  [--info]
```

Slika19: Sintaksa *neo4j-admin load* komande

--from=<archive-path> - putanja do *archive* fajla kreiranog *neo4j-admin dump* komandom na osnovu kojeg bi trebalo učitati bazu podataka, odnosno – ukoliko bi trebalo ovaj sadržaj učitati sa standardnog ulaza.

--verbose – ovom opcijom se omogućava detaljniji prikaz informacija u vezi sa izvršenjem same komande. Ovaj parametar je opcioni.

--expand-commands – opcija kojom se specificira da li će se izvršiti *extract*-ovanje same komande tokom evaluacije *config* vrednosti. Ovaj parametar je opcioni.

--database=<database> - naziv učitane baze podataka. Podrazumevana vrednost ovog parametra je *neo4j* i on je opcioni.

--force – u slučaju da je navedena ova opcija, biće izvršena zamena postojeće baze podataka.

--info – u slučaju da je navedena ova opcija, biće pružene dodatne informacije o samom *archive* fajlu koji se učitava, poput formata tog fajla, količine bajtova i sličnog.

3.4.4. PRIMER RESTORE-OVANJA OFFLINE NEO4J BAZE PODATAKA

Na slici Slika... nalazi se komanda kojom bi se izvršilo restore-ovanje offline baze podataka. Ovom komandom može se izvršiti i restore-ovanje online baze podataka, bez problema.

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

Slika20: Primer poziva za restore-ovanje Neo4j baze podataka neo4j-admin load komandom

3.5. KOPIRANJE NEO4J BAZE PODATAKA ILI BACKUP KOPIJE NEO4J BAZE PODATAKA

Korišćenjem

neo4j-admin copy

komande može se izvršiti kopiranje i korisničke baze podataka, ali i već kreirane *backup* kopije baze podataka. Ipak, ova komanda ne može biti primenjena nad sistemskom bazom podataka.

Najčešće, ova komanda se koristi prilikom rešavanja problema nekonzistentnosti podataka, a pritom je vrlo fleksibilna, jer nudi mogućnost definisanja određenih filtra kojima bi se određeni podaci mogli eliminisati iz procesa kopiranja. Tokom izvršenja ove komande ne kopira se sama šema baze podataka, odnosno njeni indeksi i ograničenja. Ipak, ukoliko baza podataka ima striktno definisanu šemu, sama komanda će u svom izlazu ponuditi određene *Cypher* upite koje bi trebalo izvršiti ne bi li smo rekreirali upravo te indekse i ograničenja.

U *Neo4j* bazi podataka svaki entitet i relacija imaju svoj jedinstveni identifikator. Prilikom izvršavanja ove komande, jedinstveni identifikatori entiteta ostaju očuvani, dok, identifikatori relacija ne, već dobijaju nove vrednosti.

```
neo4j-admin copy [--verbose]
                [--from-database=<database>]
                [--from-path=<path>]
                [--from-path-tx=<path>]
                --to-database=<database>
                [--neo4j-home-directory=<path>]
                [--force]
                [--compact-node-store]
                [--to-format=<format>]
                [--delete-nodes-with-labels=<label>[,<label>...]]
                [--keep-only-node-properties=<label.property>[,<label.property>...]]
                [--keep-only-nodes-with-labels=<label>[,<label>...]]
                [--keep-only-relationship-properties=<relationship.property>[,<relationship.property>...]]
                [--skip-labels=<label>[,<label>...]]
                [--skip-node-properties=<label.property>[,<label.property>...]]
                [--skip-properties=<property>[,<property>...]]
                [--skip-relationship-properties=<relationship.property>[,<relationship.property>...]]
                [--skip-relationships=<relationship>[,<relationship>...]]
                [--from-pagecache=<size>]
                [--to-pagecache=<size>]
```

Slika21: Sintaksa neo4j-admin copy komande

- **--verbose** – ovom opcijom se omogućava prikaz detaljnih informacija o izvršenju same komande. Parametar je opcioni.
- **--from-database=<database>** - naziv baze podataka čije kopiranje treba izvršiti. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--from-path=<path>** - putanja na kojoj se nalazi baza podataka čije bi kopiranje trebalo izvršiti. Ova opcija se najčešće koristi za specificiranje putanje na kojoj se nalaze *backup* kopije baza podataka čije bi kopiranje trebalo izvršiti. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--from-path-tx=<path>** - putanja na kojoj se nalaze *transaction log* fajlovi. Uglavnom, ovu opciju nije neophodno navoditi, međutim, ukoliko, iz nekog razloga, komanda ne može da ustanovi na kojoj se lokaciji ovi fajlovi nalaze, potrebno je specificirati putanju ovom opcijom. Parametar je opcioni i nema podrazumevanu vrednost.
- **--to-database=<database>** - naziv destinacione baze podataka. Ovo je jedini neopcionni parametar, koji je neophodno navesti prilikom poziva ove komande i nema podrazumevanu vrednost.
- **--neo4j-home-directory=<path>** - putanja *home* direktorijuma za bazu podataka koja će biti rezultat kopiranja. Podrazumevana vrednost ovog parametra je takva da će vrednost biti ista kao što je putanja *home* direktorijuma baze podataka koja se kopira.
- **--force** – ovom opcijom se omogućava da komanda bude izvršena i u slučaju da integritet baze podataka ne može biti verifikovan.
- **--to-format=<format>** - ovom opcijom se postavlja format nove baze podataka. Dozvoljene vrednosti ovog parametra su: *same*, *standard*, *high_limit* i *aligned*. Naime, *high_limit* format je dozvoljen isključivo u *Enterprise* verziji. Prilikom promene sa jednog formata na drugi, na primer sa *high_limit* na *standard*, ne postoji nikakva garancija da će podaci da se usklade sa tim. Podrazumevana vrednost parametra je takva da format nove baze podataka bude identičan formatu kopirane baze podataka.

- **--delete-nodes-with-labels=<label> [, <label>...]** – ovom opcijom specificira se labela ili više njih koje će biti korišćene kao parametar upoređivanja. Naime, bilo koji čvor koji ima definisanu bilo koju od labela navedenih kao vrednosti opcije, biće obrisani u procesu kopiranja. Tačnije, bilo koji takav čvor biće ignorisan u samom procesu kopiranja, odnosno neće biti kopiran. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--keep-only-node-properties=<label.property> [, <label.property>...]** – ovom opcijom definiše se skup atributa čvorova koje treba uključiti u proces kopiranja. Naime, bilo koji atribut *property* čvora koji ima labelu *label* biće uključen u proces kopiranja, dok svi preostali neće. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--keep-only-nodes-with-labels=<label> [, <label> ...]** – ovom opcijom specificira se lista labela koje će se koristiti kao parametar upoređivanja. Tačnije, svaki čvor koji sadrži labelu koja se nalazi u listi vrednosti ovog parametra biće uključen u proces kopiranja, dok će svi preostali biti ignorisani. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--keep-only-relationship-properties=<relationship.property> [, <relationship.property> ...]** – ovom opcijom se definiše skup atributa relacija koje bi trebalo uključiti u proces kopiranja. Naime, bilo koji atribut *property* relacije čija je labela *relationship*, biće uključen u proces kopiranja, dok će svi preostali biti ignorisani. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--skip-labels=<label> [, <label> ...]** – ovom opcijom definiše se skup labela koje će biti korišćene kao parametar za upoređivanje. Naime, svaki čvor ili relacija koja sadrži labelu koja se nalazi u listi vrednosti ovog parametra biće ignorisana tokom procesa kopiranja. Sve preostale biće uključene u ovom postupku. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--skip-node-properties=<label.property> [, <label.property> ...]** – ovom opcijom se definiše skup labela čvorova koje će biti korišćene kao parametar za upoređivanje. Naime, svaki čvor koji sadrži labelu koja se nalazi u listi vrednosti ovog parametra biće ignorisan tokom procesa kopiranja, dok će preostali biti uključeni u ovom postupku. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--skip-properties=<property> [, <property> ...]** – ovom opcijom se definiše skup naziva atributa koje treba ignorisati u procesu kopiranja. Naime, svaki atribut čiji je naziv u listi vrednosti ovog parametra biće ignorisan tokom kopiranja, dok će preostali biti uključeni u ovom procesu. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--skip-relationship-properties=<relationship.property> [, <relationship.property> ...]** – ovom opcijom se definišu nazivi atributa relacija koje bi trebalo isključiti iz procesa kopiranja. Svaki atribut relacije čiji se naziv nalazi u listi vrednosti ovog parametra biće ignorisan tokom procesa kopiranja, dok će preostali atributi biti uključeni u ovaj proces. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--skip-relationships=<relationship> [, <relationship> ...]** – ovom opcijom definiše se skup labela relacija koje se koriste kao parametar za upoređivanje. Naime, bilo koja relacija čija se labela nalazi u listi vrednosti ovog parametra biće ignorisana tokom

procesa kopiranja, dok će preostale biti uključene u ovom procesu. Ovaj parametar je opcioni i nema podrazumevanu vrednost.

- **--from-pagecache** – veličina *page* keša koji bi trebalo koristiti tokom procesa čitanja. Ovaj parametar je opcioni i nema podrazumevanu vrednost.
- **--to-pagecache** – veličina *page* keša koji bi trebalo koristiti tokom procesa upisa. Ovaj parametar je opcioni i nema podrazumevanu vrednost.

Poslednja dva parametra koriste se da bi se sam proces kopiranja ubrzao koliko god je to moguće. Napisano je pravilo da se **--to-pagecache** postavlja na vrednost između 1 i 2GB, dok bi **--from-pagecache** trebalo postaviti na najvišu moguću vrednost.

3.5.1. PRIMER KOPIRANJA BAZE PODATAKA

Postupak kopiranja baze podataka odvija se u par koraka:

Zaustaviti bazu podataka koju treba kopirati:

```
STOP DATABASE neo4j
```

Slika22a: Kopiranje Neo4j baze podataka: Zaustavljanje

Kopirati sadržaj baze podataka iz baze podataka u novu bazu podataka(baza podataka iz koje se kopira naziva se neo4j, dok se odredišna naziva copy):

```
bin/neo4j-admin copy --from-database=neo4j --to-database=copy
```

Slika22b: Kopiranje Neo4j baze podataka: Poziv komande neo4j-admin copy na izvršenje

Kreirati kopiranu bazu podataka:

```
CREATE DATABASE copy
```

Slika22v: Kopiranje Neo4j baze podataka: Kreiranje kopirane baze podataka

Pošto se izvrši kreiranje kopirane baze podataka, postupak je završen.

U slučaju da je kopirana baza podataka imala definisanu šemu baze podataka, odnosno definisane indekse i ograničenja, potrebno je rekreirati ih na osnovu komandi koje bi *neo4j-admin copy* komanda vratila kao *output*. Ovaj *output* dostupan je u *log* fajlu:

logs/neo4j-admin-copy-<timestamp>.log

4. ZAKLJUČAK

Sušтина i svrha *backup*-a je kreiranje sigurne kopije podataka koja može da se iskoristi za oporavak originalnih podataka ukoliko dođe do nekog nepredviđenog problema. Takvi problemi mogli bi da budu posledica greške u hardveru ili softveru, ali bi mogao da ih izazove i čovek, malicioznim delovanjem na podatke. Iz ovih razloga se *backup* kopije podataka smatraju temeljem i najvažnijom infrastrukturom bilo kojeg sistema.

Neo4j sistem za upravljanje bazama podataka nudi nam veći broj načina da obavimo i *backup* i *restore*. Pritom, omogućavaju izvršenje ovih operacija i za *online* i za *offline* bazu podataka.

Ova vrsta baza podataka sama po sebi namenjena je za skladištenje velike količine duboko povezanih podataka i čak i u svojim najjednostavnijim implementacijama pruža mogućnost kreiranja ogromne količine, kako čvorova, tako i relacija. Naročito kada je reč o upotrebi ove vrste *NoSQL* baza podataka u korporativne i komercijalne svrhe, veoma je važno osigurati i bezbednost podataka, ali i garanciju da ti podaci neće biti izgubljeni ni u kom slučaju.

5. LITERATURA

- [1] - <https://neo4j.com/>
- [2] - <https://www.netapp.com/data-protection/backup-recovery/what-is-backup-recovery/>
- [3] - <https://www.peerspot.com/articles/6-advantages-of-data-backup-and-recovery>
- [4] - <https://www.hubstor.net/blog/6-reasons-must-data-backup-recovery-company/>
- [5] - <https://www.rubrik.com/insights/the-importance-of-data-backup-and-recovery>
- [6] - <https://www.veritas.com/information-center/data-backup-and-recovery>
- [7] - <https://rewind.com/blog/data-backup-important-for-business/>
- [8] - <https://www.techtarget.com/searchdatabackup/definition/backup>

6. SPISAK SLIKA

Slika1: Full backup

Slika2: Incremental backup

Slika3: Differential backup

Slika4: Full backup nad Neo4j bazom podataka

Slika5: Incremental backup nad Neo4j bazom podataka

Slika6: Sintaksa neo4j-admin backup komande

Slika7: Konfiguracija servera

Slika8: Konfiguracija bezbednosti

Slika9: Primer poziva za backup-ovanje online Neo4j baze podataka neo4j-admin backup komandom

Slika10: Primer poziva za backup-ovanje većeg broja online Neo4j baza podataka neo4j-admin backup komandom

Slika11: Sintaksa neo4j-admin prepare-restore komande

Slika12: Primer poziva neo4j-admin prepare-restore komande

Slika13: Sintaksa neo4j-admin restore komande

Slika14: Primer restore-ovanja Neo4j baze podataka neo4j-admin restore komandom

Slika15: Primer izvršenja skripte za kreiranje baze podataka nakon restore operacije u Linux-u

Slika16: Primer izvršenja skripte za kreiranje baze podataka nakon restore operacije u Windows-u

Slika17: Sintaksa neo4j-admin dump komande

Slika18: Primer poziva za backup-ovanje Neo4j baze podataka neo4j-admin dump komandom

Slika19: Sintaksa neo4j-admin load komande

Slika20: Primer poziva za restore-ovanje Neo4j baze podataka neo4j-admin load komandom

Slika21: Sintaksa neo4j-admin copy komande

Slika22a: Kopiranje Neo4j baze podataka: Zaustavljanje

Slika22b: Kopiranje Neo4j baze podataka: Poziv komande neo4j-admin copy na izvršenje

Slika22v: Kopiranje Neo4j baze podataka: Kreiranje kopirane baze podataka

