

---

*Neo4j – Backup/restore  
baze podataka*

---

Student: Anđela Sokolović, 944

## Sadržaj

1.	Uvod .....	3
2.	Načini pravljenja rezervnih kopija .....	3
2.1.	Offline kopije .....	3
2.2.	Online kopije .....	4
	Potpuno kopiranje ( <i>full backup</i> ) .....	5
	Inkrementalno kopiranje ( <i>incremental backup</i> ) .....	5
	Proces kopiranja .....	5
	Konfiguracija memorije .....	7
	Rezultat kopiranja .....	7
2.3.	Provera konzistentnosti .....	8
2.4.	Planiranje kopiranja .....	9
3.	Oporavak ( <i>restore</i> ) .....	9
3.1.	Oporavak standalone servera .....	9
3.2.	Oporavak klastera .....	9
4.	Strategije pravljenja kopija i oporavka .....	10
4.1.	Baze koje treba bekapovati .....	10
4.2.	Razmatranje memorije .....	11
4.3.	Razmatranje klastera .....	11
4.4.	Korišćenje SSL/TLS protokola prilikom kopiranja .....	12
4.5.	Dodatni fajlovi koje treba iskopirati .....	14
	SSL/TLS .....	14
	Klaster .....	14
5.	Zaključak .....	14
	Literatura .....	15

## 1. Uvod

U svakom sistemu, pa i onom najzaštićenijem, postoji opasnost od dolaska do otkaza. Arhitektura svih popularnih sistema danas bazira se na tome da će do otkaza sigurno doći, samo je pitanje kada. U otkaz spada svaki vid greške koji prouzrokuje prestanak ili nepravilan rad sistema. Ovakve greške mogu imati hardversko ili softversko poreklo, a takođe mogu biti i uticaj ljudskog faktora i nepažnje. Zbog svih tipova grešaka je izuzetno bitno imati plan i strategiju za zaštitu podataka. U svim sistemima, a naročito u sistemima koji opslužuju veliki broj korisnika i čuvaju bitne i poverljive podatke, neophodno je imati ažurne kopije podataka na različitim mestima, kako bi u slučaju nepredviđenih okolnosti sistem mogao da nastavi da radi neometeno.

U ovom radu biće reči o pravljenju rezervnih kopija i oporavku podataka u bazi Neo4j, kao i o strategijama za pravljenje ovih kopija i parametrima koje treba uzeti u obzir prilikom pravljenja jedne takve strategije. Iako će se teme *backup-a* i *restore-a* dotaći i klastera računara, oni neće biti detaljnije objašnjeni, jer su izvan opsega teme koja se obrađuje u ovom radu.

## 2. Načini pravljenja rezervnih kopija

Interna struktura Neo4j baze podataka je takva da se svi podaci iz baze čuvaju u fajlovima, koji su smešteni u istom folderu u fajl sistemu, pa se pravljenje rezervne kopije u najjednostavnijem slučaju svodi na prosto kopiranje ovog direktorijuma. Postoji dve vrste backup-a sistema: online i offline.

### 2.1. Offline kopije

Pravljenje offline kopije podrazumeva da sistem bude isključen, što znači da baza neće raditi sve dok se kopiranje ne završi. Prednost ovog načina kopiranja je što je dostupan u svakoj, pa i u Community verziji baze. Očigledna mana je prestanak rada sistema tokom pravljenja kopije, pa je ovaj način nepogodan za sisteme kojima je neophodna konstantna dostupnost.

Offline proces kopiranja se sastoji iz sledećih koraka:

- Isključiti instancu baze

Kada je baza startovana na samo jednom serveru, ovaj korak je prilično jednostavan: baza se isključuje izvršavanjem odgovarajuće skripte.

```
/opt/neo4j/bin/neo4j stop
```

Nakon stopiranja baze, potrebno je isključiti i aplikaciju koja tu bazu koristi. Potrebno je osigurati da su sve instance baze i aplikacija koje koriste bazu isključene, kako bi se sistem mogao pravilno startovati nakon pravljenja kopije. Neo4j ima mogućnost zaključivanja da isključivanje nije bilo kompletno, i ugrađeni mehanizam za pokušaj oporavka u ovakvoj situaciji, ali ovaj oporavak će izazvati sporo ponovno startovanje servera, a može izazvati i naknadne probleme. Ako je instanca baze nepravilno isključena, sledeći put kada se sistem pokrene u log fajlovima će se naći poruka upozorenja: *non clean shutdown detected*.

- Iskopirati sve Neo4j fajlove iz baze na novu lokaciju

Lokacija podataka iz baze podataka je specificirana u neo4j *properties* fajlu:

```
org.neo4j.server.database.location=/var/data/neo4jdb
```

Potrebno je iskopirati ceo direktorijum i sve njegove podfoldere na novu lokaciju. Poželjno je da lokacija ne bude na istom računaru na kome je pokrenut Neo4j server.

- Restartovati instancu baze

Nakon što je kopiranje uspešno završeno, potrebno je pokrenuti bazu ponovo i proveriti log fajlove. Komanda za pokretanje baze ima sledeći format:

```
/opt/neo4j/bin/neo4j start
```

Ako u konfiguracionim fajlovima nema prijavljenih grešaka, baza je uspešno iskopirana i može se ponovo koristiti.

## 2.2. Online kopije

Za razliku od offline kopije, online kopija se pravi dok je sistem aktivan i ne zahteva nikakvo gašenje sistema, ali je dostupna isključivo u Enterprise ediciji baze. Ovaj način kopiranja je primenljiv i na pojedinačni čvor i na klaster, i tokom čitavog trajanja procesa podaci se ne zaključavaju, što znači da se mogu koristiti i dok je kopiranje u toku. Online kopiranje nudi dve mogućnosti: *full backup* i *incremental backup*.

### Potpuno kopiranje (*full backup*)

Potpuno kopiranje obuhvata kopiranje svih fajlova baze na posebnu lokaciju. Kao što je već rečeno, u toku online kopiranja nijedan fajl nije zaključan, što znači da u kopiji može doći do nekonzistentnosti ako se fajl nakon kopiranja izmeni. Ovo se rešava ugrađenim mehanizmom za praćenje transakcija: Neo4j pamti ID svake transakcije koja je započeta u toku kopiranja fajlova. Nakon što je kopiranje završeno, izvršiće se automatska provera konzistentnosti, i u slučaju da nekonzistentnost postoji, nad kopiranim fajlovima će se izvršiti sve transakcije čiji su identifikatori zapamćeni tokom procesa kopiranja. Na ovaj način, u kopiji će se naći najažurniji podaci iz originala, zaključno sa onim podacima koji su nastali pre kraja kopiranja.

### Inkrementalno kopiranje (*incremental backup*)

Inkrementalno kopiranje se razlikuje od potpunog po tome što ne kopira sve fajlove, već se bazira na pretpostavci da je najmanje jedno potpuno kopiranje obavljeno ranije. Prethodno potpuno kopiranje služi kao pokazivač na poslednju sačuvanu verziju, koji se koristi kako bi se iskopirale sve promene nastale nakon toga. Inkrementalno kopiranje prati *transaction log* i nad podacima primenjuje sve transakcije koje su se odigrale nakon čuvanja poslednjeg stanja u kopiji. Kada je u pitanju prva inkrementalna kopija, transakcije će se izvršavati u odnosu na transakcije zabeležene u potpunoj kopiji. Za sve ostale inkrementalne kopije, transakcije će se nadovezivati na transakcije iz prethodne inkrementalne kopije. Na ovaj način, pokazivač se sa svakom verzijom pomera na kraj *transaction log-a*, i svaki naredni put potrebno je iskopirati samo naknadno izvršene transakcije. Inkrementalno kopiranje je daleko efikasnije od potpunog kopiranja, jer znatno umanjuje količinu podataka koju je potrebno iskopirati, kao i vreme potrebno za sam proces kopiranja.

### Proces kopiranja

Kada se backup instanca Neo4j startuje, prvo što se proverava je da li je dozvoljeno online kopiranje (*online\_backup\_enabled*). Ovaj parametar se nalazi u konfiguracionom fajlu, i ako je verzija instance Enterprise, biće automatski omogućen. Parametar se može izmeniti ukoliko korisnik ne želi da dozvoli online kopiranje.

Ako je online kopiranje dozvoljeno, pokrenuće se poseban servis za kopiranje. Poseban servis nije neophodan, ali je preporučljiv, jer su operacije kopiranja i provere konzistencije veoma skupe i memorijski zahtevne, pa bi live server koji se koristi za uobičajene operacije

bio preopterećen. Ako je neophodno odraditi backup na istom serveru koji se koristi i u ostalim namenama, preporučuje se barem odvajanje operacija kopiranja i provere konzistentnosti, i zakazivanje ovih operacija u periodu kada je server manje opterećen. Backup servis se obično nalazi na portu 6362 kada je u pitanju pojedinačni server, ili na portu 5001 ako se radi o klasteru. Ovaj servis se koristi kao pristupna tačka na koju se konektuje alat za kopiranje i odakle se čitaju podaci i transaction log fajlovi. Ovde treba napomenuti da je alat za kopiranje zapravo Windows ili Unix skripta i može se pokrenuti sa bilo kog računara na mreži koji ima pristup servisu za kopiranje, pri čemu računar mora imati instaliran Neo4j, ali server ne mora biti pokrenut na njemu.

Komanda za backup ima sledeći format:

```
neo4j-admin backup --backup-dir=<path>
[ --verbose ]
[ --from=<host:port> ]
[ --database=<database> ]
[ --fallback-to-full=<true/false> ]
[ --pagecache=<size> ]
[ --check-consistency=<true/false> ]
[ --check-graph=<true/false> ]
[ --check-indexes=<true/false> ]
[ --check-label-scan-store=<true/false> ]
[ --check-property-owners=<true/false> ]
[ --report-dir=<path> ]
[ --additional-config=<path> ]
```

Navedene opcije imaju sledeće značenje:

- --backup-dir: Putanja do direktorijuma gde treba smestiti kopiju
- --verbose: Omogućava verbalno formirani output kopiranja
- --from: Adresa i port hosta sa kog se vrši kopiranje
- --database: Ime baze čije kopiranje se radi
- --fallback-to-full: Ako je parametar podešen na *true*, ako inkrementalno kopiranje ne uspe, sistem će premestiti staru kopiju u folder *<name>.err.<N>* i pokušati da odradi kompletno kopiranje.
- --pagecache: Definiše veličinu keša koji se može iskoristiti tokom kopiranja. Ako se ovaj parameter ne navede, podrazumevana vrednost je 8MB.
- --check-consistency: Definiše da li nakon kopiranja treba proveravati da li su kopije konzistentne.

- `--check-graph`: Definiše da li je potrebno prekontrolisati graf nakon kopiranja (proveriti čvorove, veze i attribute)
- `--check-indexes`: Definiše da li je potrebno prekontrolisati indekse nakon kopiranja
- `--check-label-scan-store`: Definiše da li treba proveravati labelle nakon kopiranja
- `--check-property-owners`: Da li je potrebno proveravati *property ownership*. Ova operacija je veoma skupa i vremenski zahtevna.
- `--report-dir`: Putanja do direktorijuma gde će biti smešten izveštaj o konzistentnosti
- `--additional-config`: Putanja do direktorijuma u kom se nalazi dodatna konfiguracija

### Konfiguracija memorije

Pre početka procesa kopiranja, moguće je definisati memoriju koja će se alocirati klijentu za kopiranje. Ovo se može uraditi dodeljivanjem vrednosti promenljivoj okruženja `HEAP_SIZE`, pre unošenja komande za početak backup-a. Ova vrednost predstavlja maksimalnu vrednost memorije koja se može dodeliti ovom procesu, a ako se ne navede, memorija će se dodeliti dinamički, shodno resursima kojima server raspolaže.

### Rezultat kopiranja

Ako destinacioni folder naveden prilikom pokretanja kopiranja nije prazan, sistem će automatski pokušati da uradi inkrementalno kopiranje (slika 1). Ako jeste, odradiće se kompletno kopiranje (slika 2):

```
$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.
```

Slika 1 - Inkrementalno kopiranje

```

$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
...
...
...

```

Slika 2 - Kompletno kopiranje

Kao rezultat kopiranja dobiće se neki od kodova:

- 0 – backup je uspešan
- 1 – backup je neuspešan
- 2 – backup je uspešan, ali je provera konzistentnosti neuspešna
- 3 – backup je uspešan, ali nije konzistentan

### 2.3. Provera konzistentnosti

Nakon što je kopiranje završeno, automatski će se pokrenuti provera konzistencije ako je dozvoljena u parametrima navedenim prilikom startovanja kopiranja. Provera konzistentnosti se mora izvršiti nad bazom koja trenutno nije u upotrebi, inače će se prijaviti greška.

Provera konzistentnosti se može izvršiti i manuelno, i to nad samom bazom ili nad kopijom, što se određuje formatom komande prilikom pokretanja:

```
neo4j-admin check-consistency [--database=<database>] | [--backup=<path>]
```

Ako je baza ili kopija konzistentna, provera će se završiti kodom 0. U slučaju da nije, provera će vratiti kod 1 i izveštaj o konzistentnosti će biti upisan u fajl

[inconsistencies-YYYY-MM-DD.HH24.MI.SS.report](#), koji će se nalaziti u folderu koji je specificiran pod `--report-dir`.

Prilikom pokretanja provere konzistentnosti, kao i prilikom kopiranja, mogu se navesti i dodatni parametri:

#### Syntax

```
neo4j-admin check-consistency [--database=<database>] | [--backup=<path>] [--verbose] [--additional-config=<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>] [--report-dir=<path>]
```

Parametri imaju isto značenje kao i parametri za pokretanje kopiranja.



## 2.4. Planiranje kopiranja

Sam Neo4j nema podršku za planiranje kopiranja, pa je za ovakvu akciju potrebno koristiti neki dodatni alat.

## 3. Oporavak (*restore*)

Kao što se proces kopiranja podataka pokazao relativno jednostavnim, tako je i sam proces oporavka veoma lak za izvođenje. Oporavak se izvršava komandom:

```
neo4j-admin restore --from=<path> [--verbose] [--database=<database>] [--force]
```

čiji parametri imaju sledeće značenje:

- `--from`: Putanja do kopije koju želimo da iskoristimo za oporavak
- `--database`: Ime baze koja se oporavlja
- `--force`: Ako baza postoji, da li je treba zameniti kopijom

### 3.1. Oporavak standalone servera

Kod samostalnog servera, koraci oporavka su veoma slični pravljenju offline kopije:

- Isključiti server na kome se nalazi Neo4j baza
- Izvršiti komandu `neo4j-admin restore` nad svakom bazom koju je potrebno oporaviti.
- Pokrenuti instancu ponovo

### 3.2. Oporavak klastera

Oporavak klastera je malo komplikovaniji, jer je potrebno ugasiti sve instance klastera i oporaviti svaku instancu ponaosob, ali je proces sličan prethodnom:

- Isključiti sve instance klastera
- Izvršiti komandu `neo4-admin unbind` na svakom serveru
- Korišćenjem komande `neo4j-admin restore`, izvršiti oporavak na svakoj instanci klastera
- Ako se oporavak vrši na novom hardveru, potrebno je obratiti posebnu pažnju na *causal settings* u konfiguracionom fajlu. Parametri o kojima je potrebno voditi računa su:

- `causal_clustering.initial_discovery_members`: lista čvorova klastera na koje je potrebno primeniti oporavak. Čvorovi se navode razdvojeni zarezom, po svojoj soket adresi.
- `causal_clustering.minimum_core_cluster_size_at_formation`: Minimalni broj čvorova koji je potreban da bi se formirao klaster. Klaster će se formirati kada je navedeni broj čvorova ispunjen, to jest kada je toliko čvorova uspeo da pronađe ostale. Podrazumevana vrednost ovog parametra je 3.
- `causal_clustering.minimum_core_cluster_size_at_runtime`: Minimalni broj čvorova koji je potreban za glasanje. Broj čvorova se može automatski menjati usled stopiranja ili startovanja čvora. Većina skupa čvorova za glasanje mora biti dostupna da bi glasanje funkcionisalo ispravno. Podrazumevana vrednost ovog parametra je 3.

## 4. Strategije pravljenja kopija i oporavka

Veoma je važno izabrati pravu strategiju za svaku bazu podataka koja učestvuje u kopiranju. Neke baze imaju veći prioritet pa im je potrebno dodeliti više resursa, dok se na drugima resursi mogu uštedeti podešavanjem određenih parametara o vremenskom zadržavanju kopije u sistemu, ili o maksimalnoj memoriji koju kopija može zauzeti. Potrebno je obratiti pažnju na ulogu koju svaka baza ima, i na osnovu toga doneti zaključak koja strategija je najpogodnija za njeno održavanje.

Prilikom formiranja strategije o pravljenju kopija i oporavaka, potrebno je voditi računa o sledećim elementima:

- Uticaj na performanse tokom kopiranja
- Tolerancija gubitka podataka u slučaju greške
- Tolerancija vremena za koje sistem neće raditi u slučaju greške
- Obim podataka

### 4.1. Baze koje treba bekapovati

Baze u Neo4j sistemu se kopiraju nezavisno jedna od druge, zbog čega je potrebno isplanirati strategiju za svaku bazu pojedinačno. Podrazumevano postoje dve baze, *neo4j* i

*system*. Sistemska baza sadrži konfiguraciju, to jest operativno stanje baza, sigurnosna podešavanja i slično. U bazi neo4j nalaze se sve korisničke baze podataka.

## 4.2. Razmatranje memorije

Za svaku kopiju, neophodno je smestiti podatke na različitom računaru od onog gde se nalazi osnovni sistem, čak je poželjno da računar bude na sasvim odvojenoj lokaciji. Ako se podaci čuvaju na cloudu, nije loše razmotriti da se kopije nalaze na odvojenom serveru, a možda čak i na cloudu kod sasvim drugog provajdera, kako bi se minimizovala šansa da se kopija ošteti kada i original. Kako se kopije čuvaju dugo, potrebno je voditi računa i o vremenu njihovog skladištenja. Na samoj kopiji se ne može uštedeti mnogo prostora, ali se deo memorije može osloboditi brisanjem transakcija koje su već primenjene iz transakcionih logova. Logovi iz transakcija se mogu izbrisati i manuelno, ali se ovo strogo ne preporučuje jer može kompromitovati podatke ukoliko se izbriše transakcija koja još uvek nije primenjena. Zbog ovoga je bolje konfigurisati maksimalnu veličinu transakcionih fajlova i prepustiti njihovo brisanje sistemu. Veličina transakcionih fajlova može se podesiti sledećim komandama:

- `dbms.tx_log.rotation.size=1M`
- `dbms.tx_log.rotation.retention_policy=3 files`.

Dodatni parametri se mogu konfigurisati kroz fajl *additional config*, na isti način koji je objašnjen ranije.

## 4.3. Razmatranje klastera

U klasteru je moguće uzeti kopiju sa bilo kog servera, pri čemu svaki server ima dva porta koji se mogu konfigurisati i služiti za kopiranje. Portovi se konfigurišu sledećim komandama:

- `dbms.backup.listen.address`
- `causal_clustering.transaction_listen_address`

Za konfiguraciju portova može se iskoristiti ista adresa za obe komande, što će pojednostaviti konfiguraciju, ali je moguće iskoristiti i dve različite adrese u slučaju da nam je potrebna određena doza fleksibilnosti.

Preporučuje se da se za backup koriste Read replike umesto Core serveri, jer ih obično ima više u sistemu i ne nose jednako opterećenje kao Core serveri. Ako dođe do pada

performansi, neka od Read replika će preuzeti ulogu one koja se trenutno koristi za backup, pa neće biti uticaja na performanse u čitavom sistemu.

Ako ne postoji pogodna Read replika koja se može iskoristiti za proces kopiranja, kopiranje se može izvršiti i na nekom od Core servera. Kriterijumi koje je poželjno razmotriti pri odabiru server su fizička udaljenost, propusnost i performanse.

Kako na klasteru može postojati više kopija istovremeno, može se dogoditi da neki čvorovi u sistemu imaju zastarelu kopiju, pa bi pre svakog procesa kopiranja trebalo proveriti ažurnost kopije na čvoru koji će biti zadužen za backup. Ova provera se može izvršiti poređenjem transakcionih fajlova glavnog čvora sa datim čvorom. Da bi se transakcioni fajlovi uporedili, dovoljno je uporediti ID poslednje izvršene transakcije, što se može učiniti `:sysinfo` komandom.

#### 4.4. Korišćenje SSL/TLS protokola prilikom kopiranja

Ako je potrebno da kopirani podaci budu enkriptovani, može se podesiti da server za kopiranje mora koristiti SSL/TLS protokol. Ako backup server zahteva neki od ovih protokla, backup klijent takođe mora biti konfigurisan na isti način. Potrebno je konfigurisati server i klijent sledećim komandama:

- `dbms.backup.listen_address` – prva dva parametra imaju isto značenje kao i pri konfiguraciji običnog backupa
- `causal_clustering.transaction_listen_address`
- `dbms.ssl.policy.backup` – potrebno je izvršiti ovu komandu i na serveru i na klijentu, i dogovoreni policy mora biti isti na oba čvora.
- `dbms.ssl.policy.cluster` – ovom komandom se dodatno definiše rad protokola na serveru

Kao što se iz navedenih primera moglo zaključiti, format komande za konfiguraciju SSL pravila je:

`dbms.ssl.policy.<scope>.<setting-suffix>`

Čitava konfiguracija se izvodi preko dva parametra, koji mogu imati različite vrednosti.

- `scope` - označava na šta se SSL pravilo primenjuje, i može uzeti vrednosti *http*, *cluster*, *backup* ili *fabric*.

- setting-suffix – sva podešavanja na koja se može uticati prilikom konfiguracije protokola. U ova podešavanja spadaju sledeća:
  - enabled – Svako pravilo se mora eksplicitno uključiti. Podrazumevana vrednost je *false*.
  - base\_directory – Osnovni direktorijum u kome se nalaze kriptografski objekti. Podrazumevana vrednost je *certificates/<scope>*.
  - private\_key – Privatni ključ koji će biti korišćen prilikom enkripcije
  - private\_key\_password – Šifra za dekodiranje privatnog ključa. Odnosi se samo na enkriptovane privatne ključeve.
  - public\_certificate – Digitalni sertifikat kome odgovara privatni ključ
  - trusted\_dir – Direktorijum u kome se nalaze sertifikati i adrese kojima se veruje
  - revoked\_dir – Direktorijum u kome se nalaze sertifikati koji više ne važe
  - verify\_hostname – Ovaj parametar se koristi za uključivanje provere imena hosta na klijentskoj strani. Nakon primanja digitalnog sertifikata od servera, klijent mora da proveri da li se adresa iz sertifikata poklapa sa poljima *common name* i *subject alternative names*. Ako se ne poklapa, veza može biti kompromitovana i klijent mora prekinuti vezu. Podrazumevana vrednost ovog parametra je *false*.
  - ciphers – Lista šifratora koja se može koristiti prilikom komunikacije.
  - tls\_versions – Lista podržanih TLS verzija
  - client\_auth – Fleg koji označava da li klijent mora proći proces autentifikacije pre početka kopiranja. Vrednosti mogu biti *none*, *optional* i *require*.
  - trust\_all – Fleg koji označava da li se svim serverima i klijentima može verovati. Ako je postavljen na *true*, sadržaj trust\_dir se ignoriše, i bilo koji server i klijent može učestvovati u komunikaciji. Podrazumevana vrednost je *false* i postavljanje ovog flega se preporučuje samo u slučaju debugiranja.

## 4.5. Dodatni fajlovi koje treba iskopirati

### SSL/TLS

Iz sigurnosnih razloga, Neo4j ne kreira foldere i fajlove vezane za SSL konfiguraciju automatski, pa tako ne učestvuje ni u njihovom backupu. Potrebno je manuelno ih napraviti, i voditi računa o njihovom kopiranju. U slučaju da se ovi fajlovi ne kopiraju manuelno, nikada se neće naći u backup planu i mogu biti izgubljeni. Kopiranje obuhvata privatne ključeve, digitalne sertifikate, i kompletne *trusted* i *revoked* direktorijume.

### Klaster

Ako se bekapovanje radi na klasteru, potrebno je voditi računa o kopiranju konfiguracione baze na svakom serveru.

## 5. Zaključak

Odlike svakog dobrog sistema su računanje na otkaz i temeljno planiranje strategije kako da se šteta izazvana tim otkazom što više umanjuje. Iz tog razloga je potrebno posvetiti dovoljno vremena planiranju održavanja kopija podataka u sistemu, i njihovog oporavka.

U radu su prikazane različite komponente o kojima treba razmišljati prilikom formiranja strategije oporavka, kao i osnovni backup metodi. Kako je svaki sistem različit i može biti razvijen na jednom ili više računara, prikazan je uporedni osvrt backup i restore metoda na standalone serverima, kao i na klasterima računara. Ono što se iz rada može zaključiti je da Neo4j obezbeđuje prilično jednostavan princip za backup i restore podataka, ali da mu nedostaje automatizacija procesa, jer je čitav raspored održavanja kopiranja upravo na odgovornosti čoveka, što može uzrokovati potencijalne probleme.

Rezervne kopije čine veliki deo dostupnosti sistema, i zato je potrebno obratiti pažnju na njih na samom početku projektovanja sistema, kako bi se za njih odvojili dovoljni resursi i isplanirao njihov detaljni razvoj, strategije i testiranje.

## Literatura

1. [Zvanična dokumentacija baze Neo4j](#)
2. Neo4j in production, Aleksa Vukotic i Nicki Watt