

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

**Neo4j – Cloud baza podataka**

Seminarski rad

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

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

Niš, jun 2022. godina

**Sadržaj**

[1. Uvod 3](#_Toc106881368)

[2. Cloud computing, DBaaS i Cloud baze podataka 3](#_Toc106881369)

[3. NoSQL Cloud baze podataka 5](#_Toc106881370)

[4. Neo4j Aura 6](#_Toc106881371)

[4.1. Primer upotrebe Neo4j AuraDB 8](#_Toc106881372)

[5. Zaključak 11](#_Toc106881373)

[6. Literatura 12](#_Toc106881374)

# Uvod

Briga oko sigurnosti podataka, kao i pristup njima u bazi podataka su veoma značajni kada je u pitanju skladištenje podataka u oblaku. Korisnik prilikom korišćenja ovakvog načina skladištenja podataka želi biti siguran da su njegovi podaci zaštićeni, a s druge strane za njega lako dostupni. Pružalac usluge skladišta, iliti Database-as-a-service, bi trebalo uvek da zna na koji način smestiti korisničke podatke u oblak, odnosno koju bazu koristiti a da odgovara skalabilnosti i distribuiranosti oblaka, kako administrirati bazu, te kome dopustiti administriranje.

U ovom radu će biti više reči o tome koje su prednosti i mane prilikom smeštanja podataka u oblak, kao i na koje sve prepreke nailaze pružaoci DaaS usluge. Isto tako na konkretnom primeru biće prikazana upotreba Neo4j baze u oblaku.

# Cloud computing, DBaaS i Cloud baze podataka

Cloud computing predstavlja globalni koncept isporuke računarskih resursa i skladišnih kapaciteta kao usluge krajnjim korisnicima i može koristiti bilo koju infrastrukturu i platformu. Oslanja se na deljenje resursa preko mreže, najčešće Interneta. Krajni korisnici pristupaju aplikacijama u oblaku preko web browser-a ili aplikacije na mobilnom telefonu, dok se softver i korisnički podaci nalaze na serverima na udaljenoj lokaciji.

Postoje tri osnovna tipa računarstva u oblaku:

* Softver kao usluga, SaaS (Software as a Service) – koji opisuje model korišćenja aplikacija na zahtev;
* Platforma kao usluga, PaaS (Platform as a Service) – koji osigurava razvojne alate za izgradnju aplikacija;
* Infrastruktura kao usluga, IaaS (Infrastructure as a Service) – koji osigurava usluge na najnižem tehničkom nivou – davanja procesorskih usluga i skladištenja podataka.

Osnovu računarstva u oblaku čini konvergentna infrastruktura, koju čine različite IT tehnologije povezane u jednu logičku i funkcionalnu celinu, kao apstrakcija fizičkih resursa vitruelizacijom, kao i deljenje resursa. U modelu računarskog oblaka razlikujemo dva odvojena dela sistema: frontend, koji je korisnički deo i obuhvata sve delove infrastrukture koji su pod kontrolom korisnika kao i sam način pristupa korisnika usluzi, i backend, koji obuhvata infrastukruru cloud provajdera. Zagovornici računarskog oblaka tvrde da ovaj model dozvoljava preduzećima da podignu i koriste aplikacije mnogo brže, sa boljom kontrolom i manje održavanja, što omogućava IT sektoru preduzeća da brže i efikasnije ispuni promenjive i nepredvidive zahteve poslovanja.

Database-as-a-Service je najkompleksnije i najmoćnije rešenje za rad i čuvanje podataka u oblaku. DBaaS nudi potpunu funkcionalnost koja se očekuje od moderne baze podataka, te je pristup moguć korišćenjem API poziva. Upravljački sloj se nalazi u pozadini i brine se o nadgledanju baze te konfigurisanju iste da bi se postigla optimalna skalabilnost, visoka dostupnost i efikasna upotreba resursa oblaka.

Da bi se usluga smatrala kao DBaaS mora zadovoljiti dva kriterijuma, a to su:

1. Usluga mora biti dostupna na zahtev, bez potrebe za instalacijom nekog softvera ili hardvera;
2. Pružalac usluge je odgovoran za održavanje baze podataka;

Na Slici 1 su prikazane neke od prednosti koriščenja DBaaS:



*Slika 1: Database as a service*

U novija vremena se sve intenzivnije koristi termin Cloud Database. Možemo ga definisati kao baze koje su prilagođene za smeštanje podataka u oblak. Baze u oblaku bi trebalo da odgovaraju na sve zahteve korisnika kao i obične baze podataka ali s druge strane, moraju i odgovarati svojstvima samog oblaka. Neke od osnovnih karaktersitika koje bi trebalo da imaju baze u oblaku su:

* visoki nivo dostupnosti;
* brzina;
* sigurnost;
* privatnost;
* pouzdanost;
* konzistentnost;
* skalabilnost;
* distribuiranost.

Posebno treba istaći da je za korisnika najbitnija visoka dostupnost baze podataka. Podaci bi trebalo krajnjem korisniku biti dostupni u svakom trenutku, dok sama replikacija podataka poboljšava pouzdanost i otpornost na greške. Ono što takođe treba naglasiti je da je radi povećanja brzine izvršavanja upita ostavljena mogućnost uključivanja paralelnog izvršavanja.

Na rad same baze podataka u oblaku ne bi trebalo da utiče njeno nadograđivanje, odnosno, dodavanje novih resursa radi poboljšanja performansi. Što nas dovodi do termina distribuirane baze podataka. Distribuirana baza podataka je skup baza smeštenih na više mašina koje su povezane na istoj mreži i prikazuju se kao jedna baza. Postoje dva tipa arhitekture ovakvih baza shared-nothing i shared-disk. Shared-nothing se bazira na podeli (particionisanju) podataka po serveru – svaki server zasebno predstavlja odvojeno skladište podataka. Ovde je problem što se dodavanjem novog servera ne poboljšavaju performanse. Shared-disk baze dopuštaju klasterima servera da koriste jednu kolekciju podataka. Svi podaci su dostupni svim serverima, nema particionisanja podataka, zahteva se manje servera, jednostavnije održavanje, visoka dostupnost.

# NoSQL Cloud baze podataka

Kao što nam je poznato od ranije nedovoljna skalabilnost relacionog DBMS-a bila je razlog pojave drugčijih mehanizama za upravljanje podacima tj. NoSQL baza podataka i MapReduce sistema. Firme poput Google, Tweeter, Facebook, Amazon koje rade s ogromnom količinom podataka kreirale su sopstvene tehnologije za skladištenje i procesiranje velike količine podataka u cloudu, nastojeći istovremeno da održe distribuiranost i skalabilnost baza. Amazon je razvio SimpleDB baziran na key-value principu (K-V baza), a Google BigTable baziranu na MapReduce okviru. Navedene baze, nazvane još i NoSQL baze podataka su zapravo čista spremišta podataka sa vrlo jednostavnim mehanizmima kontrole podataka i transakcija. Uopšteno, NoSQL baze su se pokazale pogodne za manipulaciju velikom količinom podataka u distribuiranim sistemima. Mogući nedostatak im je postojanje svojstava DBMS-a, ali one svakako nalaze primenu kod rada sa ogromnom količinom podataka. Prednost NoSQL baza je, pored rada sa ogromnom količinom podataka, i brzina i skalabilnost same baze.

U nastavku rada biće više reči o Neo4j cloud bazi podataka takozvanoj, Neo4j Auri.

# Neo4j Aura

Neo4j AuraDB je baza podataka kao usluga (Database as a service) koju hostuje Neo4j koja je dizajnirana da bude fleksibilna i veoma pouzdana u obezbeđivanju grafova za bilo koji projekat. Napravljena da iskoristi odnose u podacima, AuraDB omogućava munjevito brze upite za analitiku i uvide u realnom vremenu. Pouzdana je, sigurana i potpuno automatizovana.

Prednosti korišćenja Neo4j Aura DB:

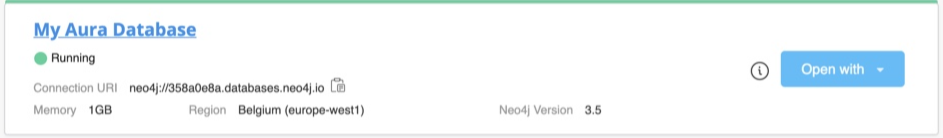
* inovativnost – same grafovske baze podataka pružaju novi način razmišljanja, a to samim tim pokreće inovacije i izvlači nas iz relacione kolotečine i skupih JOIN-a. Sam prikaz graf baze podataka u Neo4j mnogo govori o samom napretku i vizuelizaciji baze podataka kao takve;
* pouzdanost – za programere koji razvijau kritične aplikacije i oslanjanju se na cloud, Neo4j AuraDB nudi potpuno upravljani grafikon baza podataka kao brza, pouzdana i jednostavna usluga u oblaku. Neo4j AuraDB je namenski napravljen za skladištenje i analizu odnosa podataka i isporučuje munjevito brze upite za uvid u podatke u realnom vremenu. Baza podataka je automatski replicirana na dva druga sistema, stvarajući klaster koji je u potpunosti upravlja za vas. Podaci u svakom članu klastera se zatim repliciraju u dva različita data centra i automatski se distribuiraju u više zona dostupnosti.;
* laka za upotrebu – vrlo korisnički intuitivan i vizuelno primamljiv;
* skalabilnost – AuraDB automatski prilagođava i skalira bazu podataka bez ikakvih prekida ili degradacija usluge;
* brzina – sama Neo4j baza podataka skalira i vertikalno i horizontalno, bez ugrožavanja integriteta podataka, a njena arhitektura Causal Clustering podržava višestruko grupisanje. U Neo4j, odnosi podataka su prvoklasni entiteti i mogu se obići u konstantnom vremenu. Ovo omogućava da čak i složeni upiti daju rezultate u milisekundama, a ne u minutima.;
* sigurnost – sa Neo4j AuraDB podaci su šifrovani svuda: u tranzitu i u mirovanju. Sav mrežni saobraćaj, čak i unutar uslužne infrastrukture, šifrovan je pomoću HTTPS sa bezbednošću transportnog sloja (TLS). Podaci koji se čuvaju u AuraDB, uključujući i rezervne kopije, šifruju se pomoću AES-a (Advanced Encryption Standard). Neo4j AuraDB raspoređuje klastere i servisne komponente u zasebnom virtuelnom privatnom oblaku (VPC), sa namenskom infrastrukturom oblaka.;
* produktivnost – Neo4j AuraDB omogućava programerima da prave aplikacije brže i lakše sa ugrađenim drajverima, alatima i integracijama za mnoge popularne jezike i okvire;
* isplativost – Neo4j AuraDB se može pristupiti preko browser-a pa nema dodatnog instaliranja softvera isto tako sva povezivanja sa aplikacijom su vrlo laka i nema komplikovanih podešavanja.

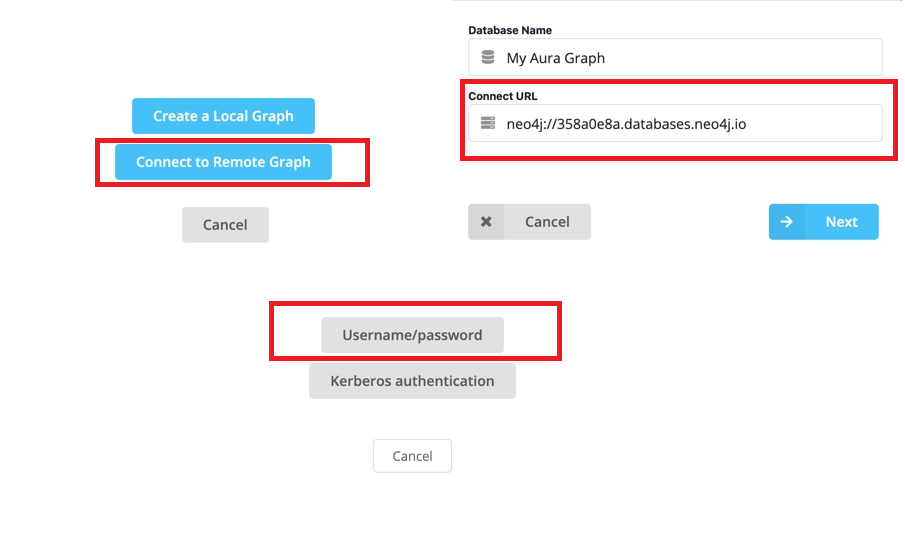
Osim ovih prednosti koje smo naveli, AuraDB ima i svoje nedostatke koji se najviše odnose na upload postojećih baza podataka na cloud. Ovo iz razloga što postoji veliki broj bezbedonsonih mehanizama koji omogućavaju da cloud baza podataka bude sigurna sama po sebi i kako bi se sprečili nedozvoljeni upadi u nju. Nedostaci su:

* direktorijum za uvoz je nedostupan - ograničavanjem pristupa internim Neo4j fasciklama zabranjuje se pristup neovlašćenim korisnicima. Učitavanje je moguće samo putem APOC[[1]](#footnote-1) procedure učitavanja ili LOAD CSV na javno dostupnim podacima;
* upotreba neo4j-admin alata za upload podataka je zabranjena - pristup disku je zaključan da bi se izbegle zlonamerne aktivnosti i oštećenje podataka/instance od strane neovlašćenih korisnika;
* uvoz lokalnih datoteka nije moguć pomoću APOC-a - Neo4j AuraDB neće pristupiti lokalnom skladištu iz zaštitnih razloga. Još uvek se može koristiti APOC za pristup javno dostupnim datotekama ili za učitavanje bezbednih datoteka;
* čuvanje API akreditiva u konfiguracionoj datoteci nije mogoće - u drugim tipovima instalacija, korisnici mogu da postave API akreditive u neo4j.conf datoteku i da ih referenciraju u Cypher upitima. Sa Neo4j AuraDB, međutim, te interne fascikle i datoteke su skrivene, tako da se ovom funkcijom mora rukovati na druge načine.

## Primer upotrebe Neo4j AuraDB

Sada kad smo se upoznali sa Neo4j Aurom i utvrdili sve njene prednosti vreme je da na primeru pokažemo korišćenje cloud baze podataka. Ono što je najpre potrebno kreirati jeste sama instanca baze podataka, najlakše je to uraditi preko Neo4j Aura konzole iz browsera, na šta posebno treba obratiti pažnju prilikom kreiranja instance je izbor regiona, odnosno fizička lokacija same instance. Što je region bliži vašoj lokaciji, brže je vreme odgovora za bilo kakve mrežne interakcije sa instancom. Jednom kada se kreira instanca, za nju dobijamo jedinstveni URL putem koga joj možemo pristupiti ukcavanjem username-a i password-a za nalog. Na Slici 2 je prikazan izgled kreirane instance u AuraDB i način konekcije na nju:



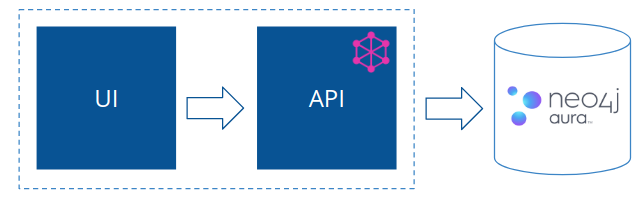


*Slika 2: AuraDB instanca baze podataka i konekcija*

Kada je kreirana instanca baze podataka vreme je osposobiti manipulaciju njenim podacima kroz neku aplikaciju: Najčešće se sve aplikacije za manipulaciju podacima sastoje od dva dela a to su:

1. API – za konekciju sa bazom i dovlačenje podataka;
2. UI – za prikaz samih podataka.

Na Slici 3 je grafički prikazano kako to zapravo izgleda:

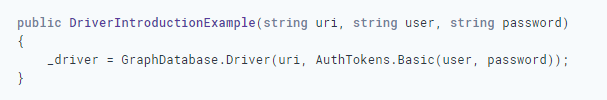


*Slika 3: Arhitektura aplikacije koja koristi Neo4j AuraDB*

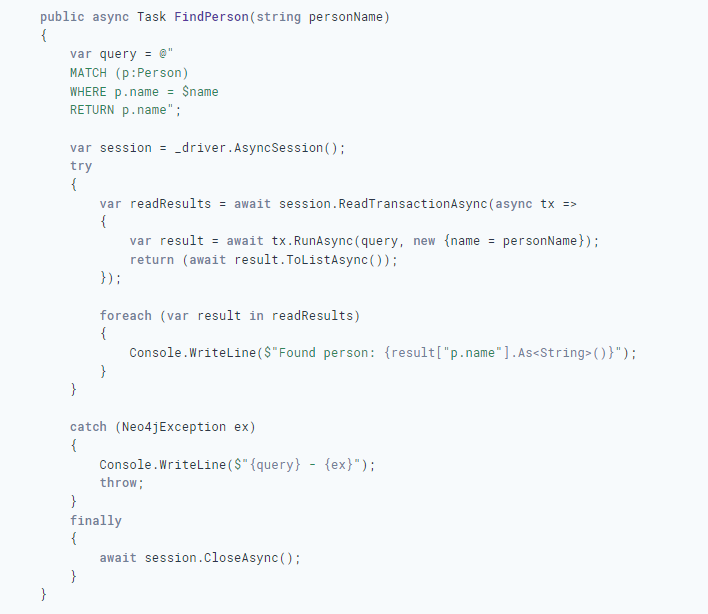
Ono što je potrebno uraditi u kodu da bi se dobila konekcija na bazu podataka, to je da se definišu parametri konekcije koje smo koristili i za logovanje na bazu, a to su URL, username i password, na primeru niže je prikazano kako to izgleda kada se koristi C#:







Prvo što primećujemo ovde je da pre samog URL-a imamo prefiks neo4j+s. To je iz razloga što Aura upiti koriste šifrovanu vezu pomoću protokola „neo4j+s“. Takođe, treba naglasiti da je pre početka rada uopšte sa Neo4j bazom podataka u C# potrebno instalirati Nuget paket Neo4j.Driver koji je zvanično podržan od Neo4j-a i povezuje aplikaciju sa bazom podataka koristeći binarni protokol. Sada kad smo se povezali na cloud bazu podataka upite iz C# koda možemo postavljati klasično kao i kada je u pitanju obična Neo4j baza, ispod je dat primer pronalaženja osobe u bazi sa zadatim imenom preko parametara funkcije:



Sa Aurom je od suštinskog značaja da se uzme u obzir najbolji region za lokaciju oblaka jer je fizička udaljenost direktan faktor u dostižnoj mrežnoj latenciji. Kada neki događaj izazove bilo kakav prekid mreže između aplikacije i Aure, na odziv bi uticalo kašnjenje mreže povratnog puta kako bi opet bilo moguće poslati upit.

# Zaključak

Cloud computing je, iako baziran na starim principima udaljenog pristupa podacima, ipak uveo jedan nov način razmišljanja, gledan iz šireg konteksta. Kod ponude usluga skladištenja podataka u oblaku, u krajnjem fokusu je korisnik kojem je potrebno osigurati privatnost podataka, zaštitu od neovlašćenog pristupa, integritet podataka, dostupnost, brz pristup podacima, gde god se oni nalazili. Ovim se korisnik lišava troškova koji se odnose na obezbeđivanje dodatnog fizičkog prostora za skladištenje velike količine podataka.

U doglednoj budućnosti može se očekivati dalji razvoj usluge Database-as-a-Service. Usluga bi se s obzirom na sigurnost i količinu podataka mogla nuditi u većim data centrima, bilo da je zakupljuju drugi pružaoci usluga ili korisnici direktno.

# Literatura

[1] Računarstvo u oblaku

[https://sr.wikipedia.org/wiki/Ra%C4%8Dunarstvo\_u\_oblaku](https://sr.wikipedia.org/wiki/Ra%C4%8Dunarstvo_u_oblaku%20)

[2] Cloud computing: što s bazom podataka (u oblacima)?

[https://bib.irb.hr/datoteka/516728.CloudComputing\_DavidovicKukuljanPogarcic.pdf](https://bib.irb.hr/datoteka/516728.CloudComputing_DavidovicKukuljanPogarcic.pdf%20)

[3] Cloud database

<https://en.wikipedia.org/wiki/Cloud_database>

[4] Cloud database Database as a service, Waleed Al Shehri, Department of Computing, Macquarie University Sydney, NSW 2109, Australia

[5] Neo4j AuraDB

[https://docs-neo4j.netlify.app/developer/aura-cloud-dbaas/](https://docs-neo4j.netlify.app/developer/aura-cloud-dbaas/%20)

[6] Neo4j Docs AuraDB

[https://neo4j.com/docs/aura/auradb/](https://neo4j.com/docs/aura/auradb/%20)

[7] Performance tuning with Neo4j AuraDB

[https://aura.support.neo4j.com/hc/en-us/articles/4404022359443-Performance-tuning-with-Neo4j-AuraDB](https://aura.support.neo4j.com/hc/en-us/articles/4404022359443-Performance-tuning-with-Neo4j-AuraDB%20)

1. APOC - biblioteka dodataka za Neo4j koja pruža stotine procedura i funkcija dodajući mnogo korisnih funkcionalnosti [↑](#footnote-ref-1)