

Seminarski rad

- Sistemi za upravljanje bazama podataka -

Tema: *Replikacija u bazi podataka MongoDB*

Student:
Marko Oblak

Mentor:
Aleksandar Stanimirović

Sadržaj

Uvod	1
-Document oriented baze podataka-	1
-NoSQL-	3
Replikacija	3
-Replikacija inače-	3
-Replikacija baze podataka MongoDB-.....	4
-Proces kreiranja skupa replika-	7
-Proces glasanja za novog primarnog čvora-	12
Zaključak.....	14
Literatura.....	15
Slike	15

Uvod

MongoDB je vodeća NoSQL baza podataka i pripada dokument orjentisanim bazama podataka (engl. document oriented). Napisana je u C++ jeziku i otvorenog je koda, izdata pod kombinacijom GNU Affero General Public License i Apache License. MongoDB čuva podatke kao JSON dokumente sa dinamičkim šemama. JSON (JavaScript Object Notation) je otvoreni standard zasnovan na tekstu, osmišljen za razmenu podataka koji su pogodni za čitanje ljudima. MongoDB je prihvaćen kao backend softver brojnih značajnih veb-sajtova i servisa, uključujući Craigslist, eBay, Foursquare, SourceForge i New York Times.

Osnovne karakteristike:

- smeštanje je usmereno na dokumenta
- apsolutna podrška indeksiranju
- automatsko skaliranje – horizontalno skaliranje bez ugrožavanja funkcionalnosti
- moćni upiti bazirani na dokumentima
- brzi update
- fleksibilna agregacija i obrada podataka
- GridFS – čuva fajlove bilo koje veličine

-Document oriented baze podataka-

Dokument orjentisane baze podataka predstavljaju softver koji se koristi za snimanje, preuzimanje i modifikovanje polu-strukturiranih podataka, dokumenata. “Document store” baze su zapravo key/ value baze u kojima je value dokument.

Dokument store znaci da:

- Objekti (podaci) se čuvaju kao dokumenti. Tako da ne postoji ‘object-relational impedance mismatch’.

- Dokumenti mogu da budu jako kompleksni. Čitav objektni model može da se učta odjednom, nema potrebe za velikim brojem CRUD atomičnih operacija.
- Dokumenti su nezavisni. Povećavaju se preformanse i smanjuju problemi za konkurentan pristup podacima.
- Otvorenog su formata. Najčešće se koriste neki široko korišćeni formati kao što su XML, JSON, BSON...
- Ne postoji striktna šema čuvanja podataka. Povećava se fleksibilnost šeme. Tako se mogu dodavati novi podaci novog formata.
- Čuvanje starih verzija. Neke baze podataka čuvaju informacije o prethodnim verzijama dokumenta u toj bazi.

Većina ovih baza obezbeđuje RESTFul api, kao i http servere koji podržavaju standardne http zahteve (put, get, post, delete).

Pretraživanje: Osim pristupa dokumentima pomocu ključa, one sadrže API pomoću kog je moguće detaljnije pretraživati bazu po nekim atributima objekata. Ne postoji standardni jezik za pretragu.

Za organizaciju dokumenata koristi se:

1. Kolekcije
2. Mehanizam tagova
3. Hierarhije direktorijuma
4. Metapodaci

Jedna od najosnovnijih karakteristika dokument store baza podataka je eventual consistency. Tako da se povremeno javlja nekonzistentnost podataka. Na račun toga ove baze podataka imaju povećanu skalabilnost i pristup podacima.

Koriste se za:

- Dinamičke podatke - CMS (Content Management Systems) i CRM (Customer Relationship Management) objekte koje korisnici mogu da menjaju i prilagođavaju sopstvenim potrebama.
- Polustrukturirane podatke
- Web podatke – sesije, logovi, shoppin cart i drugi podaci koji se održavaju za potrebe Web aplikacija. Document stores omogućavaju da se podacima pristupa kao celini jednim zahtevom ka udaljenom serveru.
- Obradu velikih količina podataka – dobra skalabilnost document store rešenja i podrška za distribuiranu obradu.

Problemi:

- Kada je potrebno koristiti transakcije (nema podrške za ACID svojstva)
- Kada je potrebno koristiti SQL upite.

-NoSQL-

Termin se upotrebljava za sve nerelacione baze podataka (non-RDBMS). Primenuju se za skladištenje velike količine podataka (distribuiranja arhitektura) – Google, Amazon, Facebook (10k – 100k servera), kada je potreba za upitima velika (kod RDBMS mala efikasnost). Pozitivne karakteristike su što su fleksibilne, skalabilne, jeftine (osnova, infrastruktura je skupa), prilagođene potrebama Web aplikacija, a loše strane su da tehnologija još uvek nije stabilna, ne postoje zajednički standardi, loša podrška za transakcije, loša podrška za pretraživanje podataka, zahteva promenu načina razmišljanja i takođe, vrlo je teško naći dva identična scenarija primene.

Replikacija

-Replikacija inače-

Replikacija je proces sinhronizacije podataka na više servera. Replikacija obezbeđuje redundantnost i povećava raspoloživost podataka sa višestrukih kopija podataka na različitim serverima baza podataka. Replikacija štiti bazu podataka od gubitka jednog servera. Takođe omogućava oporavak od hardverskog pada ili prekida usluge. Ove dodatne kopije podataka se mogu iskoristiti za oporavak od katastrofe, za izveštavanje, kao i za backup.

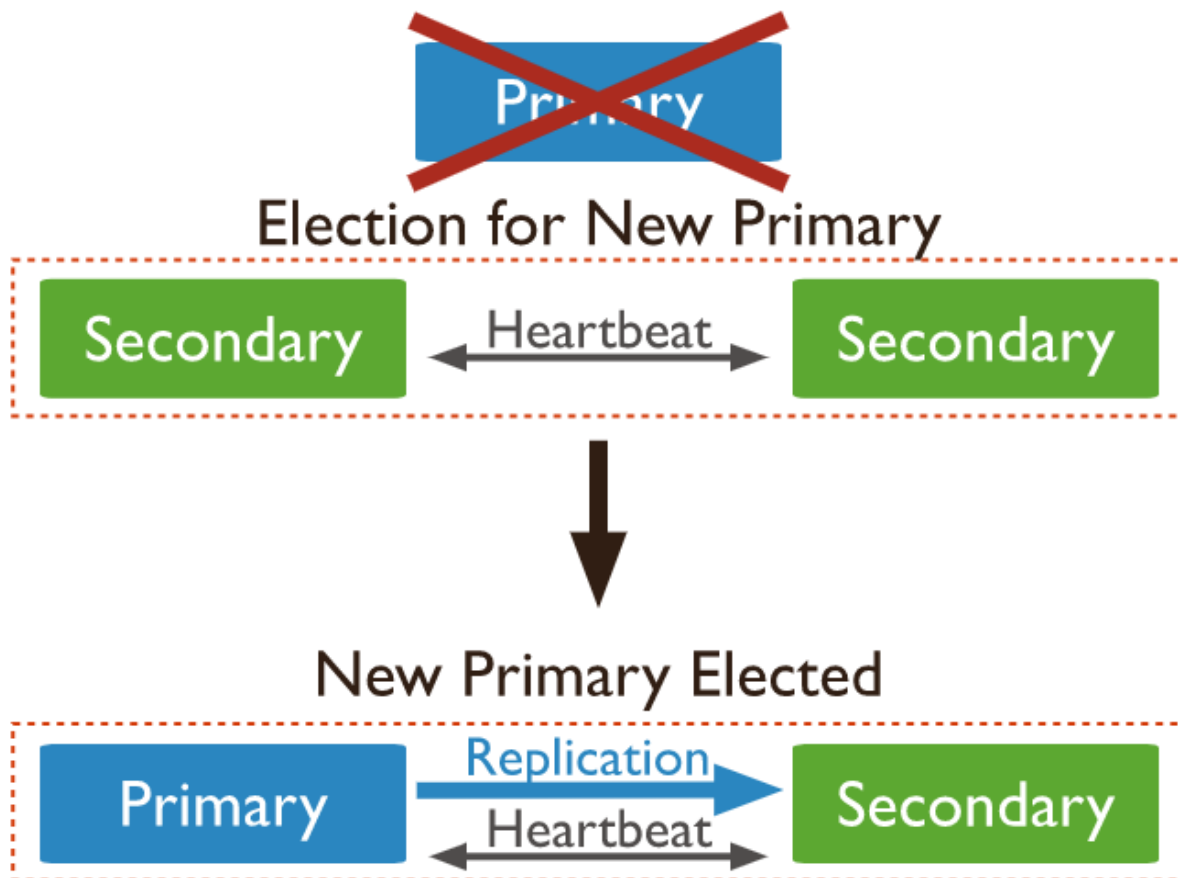
Postoji više razloga za korišćenje replikacije baze podataka:

- držati podatke u sigurnom stanju
- velika dostupnost podataka – 24/7
- oporavak od nenadanog događaja koji izaziva određene gubitke
- skalirano čitanje (postoje extra kopije sa kojih se može čitati)
- skup replika je nevidljiv aplikaciji

-Replikacija baze podataka MongoDB-

MongoDB postiže replikaciju korišćenjem skupa replika. To je grupa inicijalnih vrsta koje imaju isti skup podataka. Jedan čvor je primarni čvor i on prima sve operacije pisanja. Sve druge instance, poput sekundarnih, primenjuju operacije iz primarne i na taj način dolazi do toga da imaju isti skup podataka. Skup replika može imati samo jedan primarni čvor. Svi podaci se repliciraju iz glavnog u sekundarne čvorove.

Skup replika postavlja izbore kako bi odredila koja će instanca servera postati primarna instanca (slika 1.).



Slika 1. *Izbor novog primarnog čvora*

Otkucaji (engl. hearthbeats) – članovi skupa replika šalju takozvane otkucaje srca ili pingove ostalim članovima svake 2 sekunde. Ako se otkucaj ne vrati kroz 10 sekunde, onda se on obeležava kao nepristupačan član.

Prioritet člana – nakon što replika ima stabilnog primarnog člana, algoritam izbora će pokušati da sekundarni čvor sa najvećim prioritetom bude dostupan na izboru. Prioritet člana utiče na vreme i ishod izbora. Sekundarni čvorovi sa višim prioritetom raspisuju izbore pre nego sekundarni sa nižim prioritetom i verovatnije je da će pobediti. Međutim, instanca nižeg prioriteta se može izabrati kao primarna na kratko vreme, čak i ako je dostupna instanca sa višim prioritetom. Članovi grupe

nastavljaju sa raspisivanjem izbora sve dok član sa najvišim prioritetom ne postane primarni.

Gubitak centra podataka – pomoću distribuiranog skupa replika, gubitak centra podataka može uticati na sposobnost drugih članova u drugom centru podataka da izaberu primarnu instancu. Zato je preporučeno da se rasporede članovi skupa replika po centrima podataka kako bi se povećala verovatnoća da i uz gubitak centra podataka, jedan od preostalih članova skupa replika može postati primarni.

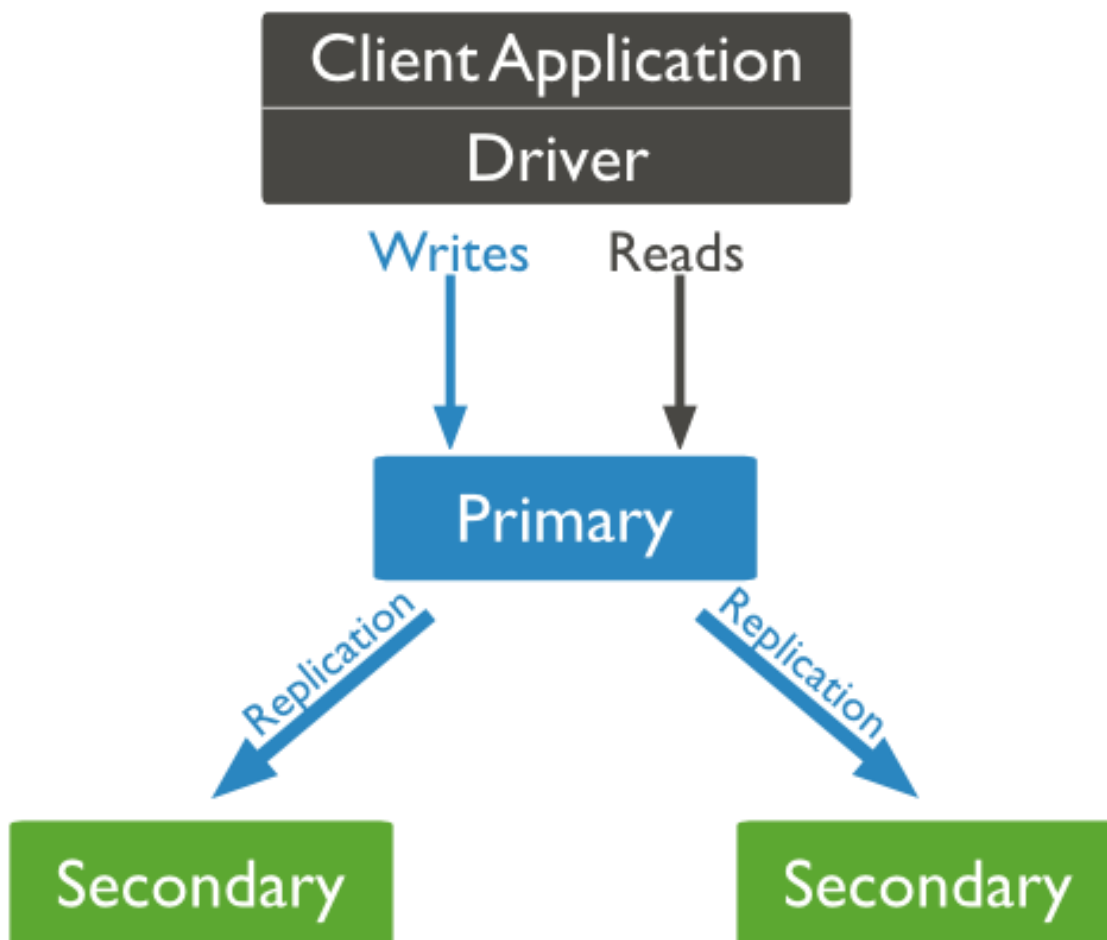
Mrežna particija - može segregirati primarni u particiju sa manjinom čvorova. Kada primarni otkrije da može videti samo manjinu čvorova u skupu replika, primarni postaje sekundarni. Nezavisno, član particije koji može komunicirati s većinom čvorova (uključujući sebe) održava izbore kako bi postao novi glavni.

Skup replika pokreće ovaj izbor kao odgovor na različite događaje:

- dodavanje nove replike u skup replika
- inicijalizacija skupa replika
- pozivanjem određenih metoda za održavanje skupa replika kao što su *rs.stepDown()* ili *rs.reconfig()*
- sekundarne instance gube povezanost sa glavnom instancom duže od konfigurisanog vremenskog ograničenja (inicijalno je to 10 sekundi)

Nakon što se čvor oporavi i ponovo vrati u skup replika nastavlja rad kao sekundarni čvor.

Tipični dijagram koji je prikazan na slici ispod (slika 2.) pokazuje da klijentska aplikacija uvek interaguje sa primarnim čvorom, a primarni čvor onda replicira podatke na sekundarne čvorove.



Slika 2. *Dijagram povezanosti skupa replika sa klijentskom aplikacijom*

-Proces kreiranja skupa replika-

Najpre se za određene čvorove kreiraju posebni folderi koji sadrže po tri fajla karakteristična i potrebna u procesu izvršenja ovih instanci – *db*, *conf* i *log* fajl.

U konfiguracionom fajlu se mogu navesti osnovne informacije prikazane na slici 3. – putanja do direktorijuma podataka, putanja do direktorijuma gde se nalazi log fajl, port na kom će ova instanca osluškivati kao i ime skupa replika.

```
# mongod.conf

# data directory
dbpath=C:\data\db

# log file
logpath=C:\data\log\mongo.log

logappend=true

#port number
port=27017

#replica set name
replSet=rs1
```

Slika 3. *Primer konfiguracionog fajla jedne instance Mongo servera*

U ovom radu će biti predstavljen rad tri instance servera Mongo sistema, pri čemu je jedna instanca uvek primarna a ostale su sekundarne. Navedena konfiguracija je specifična samo za jednu instancu, a na isti način se kreiraju i konfiguracioni fajlovi za preostale dve instance.

Sada sa definisanim instancama možemo kreirati određene servise (Windows servise) i pokrenuti ih na sledeći način (slika 4.)

```
C:\Program Files\MongoDB\Server\4.2\bin>mongod --config C:\data\config\mongod.cfg --install

C:\Program Files\MongoDB\Server\4.2\bin>net start mongodb
The MongoDB service is starting.
The MongoDB service was started successfully.

C:\Program Files\MongoDB\Server\4.2\bin>
```

Slika 4. *Primer kreiranja instance servisa i pokretanje*

Nakon što se na isti način kreiraju ostale dve instance servera i iste pokrenu, neophodno je konfigurisati i skup replika na sledeći način (slike 5., 6. i 7.)

```
C:\mongodb-2.4.4\bin>mongo --port 27017
```

Slika 5. Primer pokretanje instance servera na portu 27017

```
> config = { _id: "rs1", members:[  
... { _id : 0, host : "localhost:27017"},  
... { _id : 1, host : "localhost:37017"},  
... { _id : 2, host : "localhost:47017"}  
... ] }
```

Slika 6. Konfiguracija skupa replika

```
> rs.initiate(config)  
{  
  "info" : "Config now saved locally.  Should come online in about a minute.",  
  "ok" : 1  
}
```

Slika 7. Konfigurisanje skupa replika zadatom konfiguracijom

```

> rs.status()
{
  "set" : "rs1",
  "date" : ISODate("2013-07-02T18:40:27Z"),
  "myState" : 1,
  "members" : [
    {
      "_id" : 0,
      "name" : "localhost:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 651,
      "optime" : {
        "t" : 1372790393,
        "i" : 1
      },
      "optimeDate" : ISODate("2013-07-02T18:39:53Z"),
      "self" : true
    },
    {
      "_id" : 1,
      "name" : "localhost:37017",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 31,
      "optime" : {
        "t" : 1372790393,
        "i" : 1
      },
      "optimeDate" : ISODate("2013-07-02T18:39:53Z"),
      "lastHeartbeat" : ISODate("2013-07-02T18:40:26Z"),
      "lastHeartbeatRecv" : ISODate("1970-01-01T00:00:00Z"),
      "pingMs" : 0,
      "syncingTo" : "localhost:27017"
    },
    {
      "_id" : 2,
      "name" : "localhost:47017",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 31,
      "optime" : {
        "t" : 1372790393,
        "i" : 1
      },
      "optimeDate" : ISODate("2013-07-02T18:39:53Z"),
      "lastHeartbeat" : ISODate("2013-07-02T18:40:26Z"),
      "lastHeartbeatRecv" : ISODate("1970-01-01T00:00:00Z"),
      "pingMs" : 0,
      "syncingTo" : "localhost:27017"
    }
  ],
  "ok" : 1
}

```

Slika 8. Provera statusa skupa replika

Ako pokrenemo dve od tri instance servera jasno se vidi koja je instanca primarna a koja je sekundarna (slika 9. i 10.)

```
C:\Program Files\MongoDB\Server\4.2\bin>mongo --port 37017
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:37017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("59349ec2-98fe-4ef7-8159-2efb35a733e7") }
MongoDB server version: 4.2.6
Server has startup warnings:
2020-05-14T17:19:53.319+0200 I CONTROL [initandlisten]
2020-05-14T17:19:53.319+0200 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-05-14T17:19:53.319+0200 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten]
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify which IP
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --bind_ip_all to
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, start the
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-05-14T17:19:53.320+0200 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
rs1:PRIMARY>
```

Slika 9. Primarna replika

```
C:\Program Files\MongoDB\Server\4.2\bin>mongo --port 27017
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0e163e92-9474-44bb-a155-b926a40e010e") }
MongoDB server version: 4.2.6
Server has startup warnings:
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten]
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten]
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server.
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify which IP
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --bind_ip_all to
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, start the
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-05-14T18:18:41.427+0200 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
rs1:SECONDARY>
```

Slika 10. Sekundarna replika

U nastavku ćemo proveriti ispravnost rada skupa replika.

Sledeće slike pokazuju proces upisa podataka korišćenjem primarne instance, nakon čega iste podatke čitamo pomoću sekundarne instance (slika 11. i slika 12.)

```
rs1:PRIMARY> use portraitGallery
switched to db portraitGallery
rs1:PRIMARY> db.person.save({"name":"dbPersonPro"})
WriteResult({ "nInserted" : 1 })
rs1:PRIMARY> █
```

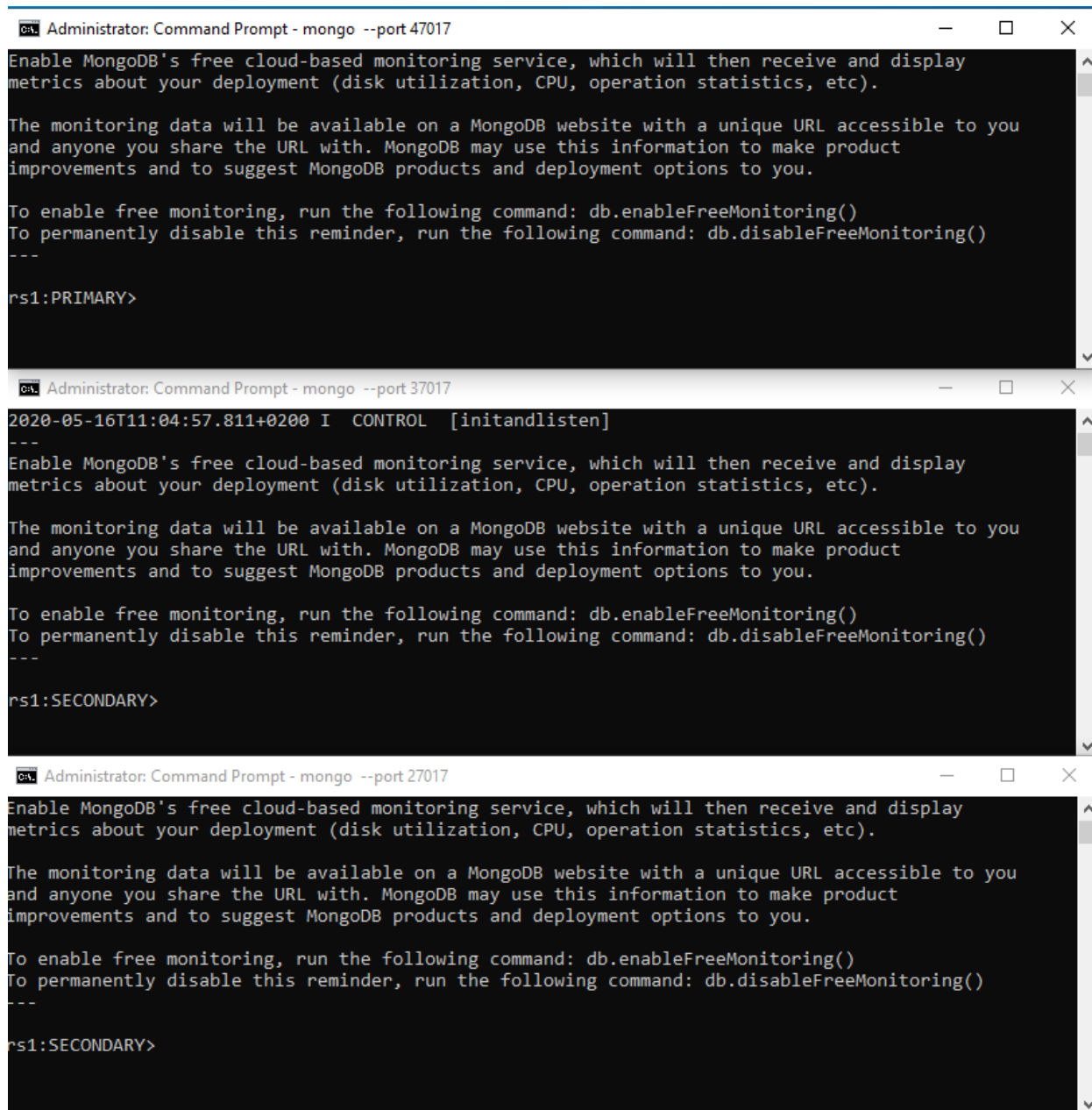
Slika 11. *Upis podataka glavne replike*

```
rs1:SECONDARY> rs.slaveOk()
rs1:SECONDARY> use portraitGallery
switched to db portraitGallery
rs1:SECONDARY> db.person.find()
{ "_id" : ObjectId("5ebd072503d782da60fe7597"), "name" : "Mareeee" }
{ "_id" : ObjectId("5ebd07f803d782da60fe7598"), "name" : "Milica" }
{ "_id" : ObjectId("5ebd76aa82c778fbf4990858") }
{ "_id" : ObjectId("5ebd771865376c2210adc1cf"), "name" : "dbPersonPro" }
rs1:SECONDARY>
```

Slika 12. *Čitanje podataka sekundarne replike*

-Proces glasanja za novi primarni čvor-

Ako pokrenemo tri replike, videćemo da je jedna obeležena kao primarna, a ostale dve kao sekundarne replike. U ovom slučaju je primarna replika ona koja osluškuje na portu 47017. Ako simuliramo gašenje servisa glavne replike tako što ručno prekinemo status izvršavanja, videćemo da nakon vrlo kratkog vremena neka druga sekundarna replika dobija ulogu primarne (u ovom slučaju instanca servera koja osluškuje na portu 27017). Na sledećim slikama (13. i 14.) praktično je prikazan ovaj proces.



Slika 13. *Prikaz pokrenutih replika*

The image consists of two screenshots of a Windows Command Prompt window titled "Administrator: Command Prompt - mongo --port 37017".

The top screenshot shows the following text:

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

rs1:SECONDARY> _

The bottom screenshot shows the following text:
2020-05-16T11:04:57.814+0200 I CONTROL [initandlisten]

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

rs1:PRIMARY>

Slika 14. *Prikaz rezultata glasanja za novu primarnu repliku*

Zaključak

Sve u svemu, može se reći da je repliciranje MongoDB baze podataka vrlo jednostavno i pomaže u izgradnji distribuiranih aplikacija. Replika baze podataka se može rasporediti na mnogim serverima. Dakle, ako server padne, podaci se mogu preuzeti i sa drugih servera i to neće oštetiti funkcionisanje baze podataka.

Literatura

<https://www.percona.com/community-blog/2019/01/14/introduction-mongodb-replication/>

<https://docs.mongodb.com/manual/core/replica-set-elections/>

<https://malderhout.wordpress.com/2012/11/02/implement-mongodb-replication-in-3-simple-steps/>

<https://www.guru99.com/mongodb-atlas-cloud.html>

https://www.tutorialspoint.com/mongodb/mongodb_replication.htm

Slike

Slika 1. Izbor novog primarnog čvora

Slika 2. Dijagram povezanosti skupa replika sa klijentskom aplikacijom

Slika 3. Primer konfiguracionog fajla jedne instance Mongo servera

Slika 4. Primer kreiranja instance servisa i pokretanje

Slika 5. Primer pokretanje instance servera na portu 27017

Slika 6. Konfiguracija skupa replika

Slika 7. Konfigurisanje skupa replika zadatom konfiguracijom

Slika 8. Provera statusa skupa replika

Slika 9. Primarna replika

Slika 10. Sekundarna replika

Slika 11. Upis podataka glavne replike

Slika 12. Čitanje podataka sekundarne replike

Slika 13. Prikaz pokrenutih replika

Slika 14. Prikaz rezultata glasanja za novu primarnu repliku