*Elektronski fakultet Niš*



*Sistemi za upravljanje bazama podataka* **DBaaS – MongoDB**

Mentor:

prof. dr Aleksandar Stanimirović

Student:

David Bosnić 1289

Sadržaj:

[Uvod 3](#_Toc106561290)

[Usluge zasnovane na oblaku 4](#_Toc106561291)

[Vrste računarstva u oblaku 4](#_Toc106561292)

[Softver kao usluga (SaaS) 4](#_Toc106561293)

[Prednosti SaaS-a 5](#_Toc106561294)

[Platforma kao usluga 5](#_Toc106561295)

[Prednosti *PaaS*-a: 5](#_Toc106561296)

[Infrastruktura kao usluga 6](#_Toc106561297)

[Prednosti *IaaS*-a: 6](#_Toc106561298)

[Bilo šta kao usluga 6](#_Toc106561299)

[Funkcija kao usluga 7](#_Toc106561300)

[Prednosti *FaaS*-a: 7](#_Toc106561301)

[Baza podataka kao usluga 8](#_Toc106561302)

[Funkcionisanje *DBaaS*-a 8](#_Toc106561303)

[Razlika principa baze podataka kao usluge od upravljanja bazom podataka 9](#_Toc106561304)

[Osobine *DBaaS*-a 9](#_Toc106561305)

[Korišćenje hardvera kod *DBaaS*-a 9](#_Toc106561306)

[Brzina implementacije baze podataka 10](#_Toc106561307)

[Operacije sa podacima za buduće upotrebe 10](#_Toc106561308)

[Korišćenje *DBaaS*-a 10](#_Toc106561309)

[Izbor *DBaaS*-a 11](#_Toc106561310)

[Zaključak 12](#_Toc106561311)

[Reference 13](#_Toc106561312)

Uvod

Kao što je poznato, podaci su od krucijalne važnosti za sve kategorije preduzeća. Svakodnevno se u svakoj organizaciji generiše ogromna količina podataka i na osnovu ovih prikupljenih ili generisanih podataka obavljaju se različite organizacione aktivnosti. Ali najizazovniji zadatak vezan za podatke je upravljanje bazom podataka i obezbeđivanje bezbednosti baze podataka. Tako se svakodnevno nove tehnologije dodaju domenu baze podataka kako bi rad bio lakši i fleksibilniji. *DBaaS* je jedno od naprednih poboljšanja koje aktivnosti baze podataka čine jednostavnijim.

U ovom radu će biti dublje objašnjen pojam baze podataka kao usluge (*DBaaS*), kako ona funkcioniše, kako pomaže u svakodnevnom radu baze podataka i na kraju o njenim prednostima i nedostacima.

Usluge zasnovane na oblaku

Računarstvo u oblaku se može definisati kao praksa korišćenja mreže udaljenih servera hostovanih na Internetu radi skladištenja, upravljanja i obrade podataka, a ne kao lokalni server ili lični računar. Kompanije koje nude takve vrste usluga računarstva u oblaku nazivaju se dobavljačima u oblaku i obično naplaćuju usluge računarstva u oblaku na osnovu korišćenja. Mreže i klasteri su osnova za računarstvo u oblaku.

# **Vrste računarstva u oblaku**

Većina usluga računarstva u oblaku spada u par širih kategorija:

* Softver kao usluga (*SaaS*)
* Platforma kao usluga (*PaaS*)
* Infrastruktura kao usluga (*IaaS*)
* Bilo šta kao usluga (*XaaS*)
* Funkcija kao usluga (*FaaS*)

Oni se ponekad nazivaju stekom računarstva u oblaku, jer su izgrađeni jedan na drugom. Ovi slojevi apstrakcije se takođe mogu posmatrati kao slojevita arhitektura gde usluge višeg sloja mogu biti sastavljene od usluga donjeg sloja, tj. *SaaS* može da obezbedi infrastrukturu.

# **Softver kao usluga (SaaS)**

Softver kao usluga (*SaaS*) je način pružanja usluga i aplikacija preko Interneta. Umesto da se instalira i održava softver, jednostavno mu se pristupa preko Interneta, oslobađajući se složenog upravljanja softverom i hardverom. Korišćenje *SaaS*-a uklanja potrebu za instaliranjem i pokretanjem aplikacija na sopstvenim računarima ili u data centrima eliminišući troškove hardvera kao i održavanja softvera.

*SaaS* pruža kompletno softversko rešenje koje se uglavnom kupuje po principu plaćanja od dobavljačima usluga u oblaku dok traje korišćenje resursa. Većina *SaaS* aplikacija se može pokrenuti direktno iz veb pretraživača bez potrebe za preuzimanjem ili instalacijom. *SaaS* aplikacije se ponekad nazivaju softver zasnovan na vebu, softver na zahtev ili hostovani softver.

## Prednosti SaaS-a

* Isplativost: Plaćanje samo onog što se koristi.
* Smanjeno vreme: Korisnici mogu da pokreću većinu *SaaS* aplikacija direktno iz svog veb pretraživača bez potrebe za preuzimanjem i instaliranjem bilo kakvog softvera. Ovo smanjuje vreme utrošeno na instalaciju i konfiguraciju i može smanjiti probleme koji mogu da ometaju implementaciju softvera.
* Pristupačnost: Možemo pristupiti podacima aplikacije sa bilo kog mesta.
* Automatska ažuriranja: Umesto da kupuju novi softver, korisnici se oslanjaju na *SaaS* provajdera da automatski izvrši ažuriranje.
* Skalabilnost: Omogućava korisnicima da pristupe uslugama i funkcijama na zahtev.
* Različite kompanije koje pružaju softver kao uslugu su Cloud9 Analitics, Salesforce.com, Cloud Switch, Microsoft Office 365, Big Commerce, Eloqua, DropBox i Cloud Tran.

# **Platforma kao usluga**

*PaaS* je kategorija računarstva u oblaku koja obezbeđuje platformu i okruženje koje omogućava programerima da prave aplikacije i usluge preko interneta. *PaaS* usluge se hostuju u oblaku i korisnici im pristupaju jednostavno preko svog veb pretraživača.

*PaaS* provajder hostuje hardver i softver na sopstvenoj infrastrukturi. Kao rezultat toga, *PaaS* oslobađa korisnike od potrebe da instaliraju interni hardver i softver za razvoj ili pokretanje nove aplikacije. Dakle, razvoj i primena aplikacije se odvijaju nezavisno od hardvera.

Potrošač ne upravlja niti kontroliše osnovnu infrastrukturu oblaka, uključujući mrežu, servere, operativne sisteme ili skladište, ali ima kontrolu nad primenjenim aplikacijama i eventualno podešavanjima konfiguracije za okruženje za hostovanje aplikacija. Da bi bilo jednostavnije, treba uzeti za primer jednogodišnje dnevne funkcije, postoje dve opcije ili da se kreira mesto ili da se iznajmi mesto, ali funkcija je ista.

## Prednosti *PaaS*-a:

* Jednostavan i zgodan za korisnike: Pruža veliki deo infrastrukture i drugih IT usluga, kojima korisnici mogu pristupiti bilo gde preko veb pretraživača.
* Isplativo: naplaćuje usluge koje se pružaju na osnovu upotrebe, čime se eliminišu troškovi koji se mogu imati za hardver i softver na lokalnom nivou.
* Efikasno upravljanje životnim ciklusom: Dizajniran je da podrži kompletan životni ciklus veb aplikacije: izgradnju, testiranje, primenu, upravljanje i ažuriranje.
* Efikasnost: Omogućava programiranje višeg nivoa sa smanjenom složenošću, tako da ukupni razvoj aplikacije može biti efikasniji.
* Različite kompanije koje pružaju Platformu kao uslugu su Amazon veb servisi Elastic Beanstalk, Salesforce, Windows Azure, Google App Engine, Cloud Bess i IBM pametni oblak.

# **Infrastruktura kao usluga**

Infrastruktura kao usluga (*IaaS*) je model usluge koji isporučuje računarsku infrastrukturu na bazi spoljnih saradnika za podršku različitim operacijama. Tipično *IaaS* je usluga u kojoj se infrastruktura obezbeđuje kao spoljni izvor preduzećima kao što su mrežna oprema, uređaji, baza podataka i veb serveri.

Takođe je poznat kao Hardver kao usluga (*HaaS*). *IaaS* klijenti plaćaju po korisniku, obično po satu, nedelji ili mesecu. Neki provajderi takođe naplaćuju klijentima na osnovu količine prostora virtuelne mašine koju koriste.

On jednostavno obezbeđuje osnovne operativne sisteme, bezbednost, umrežavanje i servere za razvoj takvih aplikacija i usluga, i primenu razvojnih alata, baza podataka itd.

## Prednosti *IaaS*-a:

* Isplativost: Eliminiše kapitalne troškove i smanjuje tekuće troškove, a *IaaS* klijenti plaćaju po korisniku, obično po satu, nedelji ili mesecu.
* Hostovanje veb lokacija: Pokretanje veb lokacija koje koriste *IaaS* može biti jeftinije od tradicionalnog veb hostinga.
* Bezbednost: *IaaS* Cloud Provider može da obezbedi bolju bezbednost od postojećeg softvera.
* Održavanje: Nema potrebe za upravljanjem osnovnim centrom podataka ili uvođenjem novih izdanja razvojnog ili osnovnog softvera. Sve ovo obavlja *IaaS* Cloud Provider.
* Različite kompanije koje pružaju infrastrukturu kao uslugu su Amazon veb usluge, Bluestack, IBM, Openstack, Rackspace i Vmware.

# **Bilo šta kao usluga**

Većina provajdera usluga u oblaku danas nudi bilo šta kao uslugu koja je kompilacija svih gore navedenih usluga, uključujući neke dodatne usluge.

Prednosti *XaaS*-a: Pošto je ovo kombinovana usluga, ima sve prednosti svake vrste usluge u oblaku.

# **Funkcija kao usluga**

*FaaS* je vrsta usluge računarstva u oblaku. Svojim korisnicima ili klijentima pruža platformu za razvoj, izračunavanje, pokretanje i primenu koda ili cele aplikacije kao funkcije. Omogućava korisniku da u potpunosti razvije kod i ažurira ga u bilo kom trenutku bez brige o održavanju osnovne infrastrukture. Razvijeni kod se može izvršiti uz odgovor na određeni događaj. Takođe je isto što i *PaaS*.

*FaaS* je model izvršenja vođen događajima. Implementiran je u kontejneru bez servera. Kada se aplikacija u potpunosti razvije, korisnik će pokrenuti događaj da izvrši kod. Sada, pokrenuti događaj daje odgovor i aktivira servere da ga izvrše. Serveri nisu ništa drugo do Linux serveri ili bilo koji drugi serveri kojima u potpunosti upravlja prodavac. Kupac nema pojma ni o jednom serveru, zbog čega ne moraju da održavaju server, stoga je arhitektura bez servera.

I *PaaS* i *FaaS* pružaju istu funkcionalnost, ali još uvek postoji određena razlika u pogledu skalabilnosti i cene.

*FaaS*, obezbeđuje automatsko povećanje i smanjenje u zavisnosti od potražnje. *PaaS* takođe obezbeđuje skalabilnost, ali ovde korisnici moraju da konfigurišu parametar skaliranja u zavisnosti od potražnje.

U *FaaS*-u, korisnici moraju da plate samo za broj vremena izvršenja. U *PaaS*-u, korisnici moraju da plate iznos na osnovu cene koja se plaća bez obzira na to koliko koriste.

## Prednosti *FaaS*-a:

* Visoko skalabilan: Automatsko skaliranje vrši provajder u zavisnosti od potražnje.
* Isplativost: Plaća se za broj izvršenih događaja.
* Pojednostavljenje koda: *FaaS* omogućava korisnicima da otpreme celu aplikaciju odjednom. Omogućava da se kod piše za nezavisne funkcije ili slične njima.
* Održavanje koda je dovoljno i nema potrebe za brigom o serverima.
* Funkcije se mogu napisati na bilo kom programskom jeziku.
* Manja kontrola nad sistemom.
* Različite kompanije koje pružaju funkciju kao uslugu su Amazon Web Services – Firecracker, Google – Kubernetes, Oracle – Fn, Apache Openwhisk – IBM, OpenFaaS,

Baza podataka kao usluga

Baza podataka kao usluga (*DBaaS*) je usluga računarstva u oblaku. Kao hostovana/upravljana usluga, korisnici ne moraju da brinu o postavljanju hardvera ili instaliranju softvera. Sve što je u vezi sa upravljanjem bazom podataka obavlja provajder usluga.

Opcije hostovanja baze podataka su dostupne za sve tipove baza podataka, uključujući *NoSQL*, *MySQL* i *PostgreSQL*. *MongoDB* *Atlas* je jedan primer *NoSQL* *DBaaS* servisa koji je lako skalabilan.

*DBaaS* pretplata uključuje sve što je potrebno za rad baze podataka u oblaku – uključujući obezbeđivanje baze podataka, licence, podršku i održavanje. Programeri mogu da iskoriste API-je hostovane u oblaku za pravljenje novih aplikacija, programski pristupajući podacima i manipulišući njima. Zbog toga, *DBaaS* ima mnogo sličnosti sa drugim ponudama u oblaku zasnovanim na *SaaS* pretplati.

Kao upravljana usluga, nema dodatnih troškova, može se odmah početi sa manipulisanjem podacima u okviru baza podataka.

*DBaaS* je logično proširenje *cloud* tehnologija, koristeći objedinjene kapacitete za skladištenje i obradu kako bi podržao promenljive zahteve korisnika platforme. Sa praktično beskonačnim obimom rasta podataka, računarstvo u oblaku pomaže da se prevaziđu fizička ograničenja lokalnog centra podataka.

# **Funkcionisanje *DBaaS*-a**

Kada se podaci učitaju, sam *DBaaS* mehanizam baze podataka radi na skoro potpuno isti način kao i lokalna instalacija. U stvari, isto jezgro je instalirano u hostovanom data centru. Za programere i inženjere podataka, iskustvo se gotovo ne razlikuje od rada sa lokalnom bazom podataka.

Glavna razlika je fizička infrastruktura na kojoj radi baza podataka u oblaku. U javnom okruženju u oblaku „Infrastruktura kao usluga“ (*IaaS*), kao što su *Microsoft Azure*, *Amazon* *Web Services* (*AWS*), *Google Cloud Platform (GCP)* i *MongoDB* Atlas, mehanizam baze podataka (i podaci) se pokreće na zajedničkoj hardverskoj platformi. Ovo dodaje računarsku snagu, elastičnost resursa i skalabilnost potrebne za podršku rastućih skladišta podataka i potreba obrade.

## Razlika principa baze podataka kao usluge od upravljanja bazom podataka

Upravljanje bazom podataka u oblaku je često mnogo jednostavnije od tradicionalnih lokalnih ekvivalenata. Sami alati za administraciju baze podataka su skoro identični, omogućavajući da se veoma brzo i lako obezbedi baza podataka na hostovanoj infrastrukturi.

Glavna razlika između *DBaaS*-a i lokalnih implementacija je količina potrebne pozadinske administracije. Principi računarstva u oblaku omogućavaju da se prenese dugotrajna administracija infrastrukture na dobavljača usluga. Oni su odgovorni da obezbede da fizički i aplikativni slojevi budu operativni i optimizovani.

Bez obzira da li sistem koristi individualni programer ili tim inženjera podataka i programera, administracija infrastrukture za spoljne poslove obezbeđuje da se rad fokusira na same podatke, vraćajući vreme i resurse koji bi inače bili potrošeni na zadatke održavanja niskog nivoa.

# **Osobine *DBaaS*-a**

Prednosti, a ujedno i razlozi zbog čega treba koristiti princip *DBaaS*-a se mogu grupisati u tri celine:

* Nema dodatnog hardvera
* Brza implementacija baze podataka
* Operacije sa podacima spremne za budućnost

## Korišćenje hardvera kod *DBaaS*-a

Veličine baza podataka nastavljaju eksponencijalno da rastu, jer moderne aplikacije sada indeksiraju, pretražuju i obrađuju širok spektar tipova datoteka, uključujući video, audio i druge nestruktuirane formate. Ovo zahteva redovno ulaganje u dodatne kapacitete za skladištenje i obradu kako bi se omogućilo proširenje.

Sa *DBaaS*-om, model računarstva u oblaku nudi praktično neograničen potencijal rasta bez ikakvih unapred ulaganja. Baza podataka može nastaviti da raste bez brige oko dostizanja kapaciteta ili potrebe za ulaganjem unapred u dodatni hardver.

*DBaaS* uslugama se takođe u potpunosti upravlja, tako da provajderi (kao što je *MongoDB*) brinu o infrastrukturi, hardveru, operativnom sistemu i softveru. Ovo oslobađa vreme i programere/analitičare podataka da se fokusiraju na izgradnju aplikacija ili na izvlačenje vrednosti iz baze podataka, bez dodatnih resursa potrebnih za upravljanje i održavanje platforme.

## Brzina implementacije baze podataka

Korišćenjem baze podataka kao modela usluge, proširenje i poboljšanje operacija sa podacima je znatno pojednostavljeno. Problem dostupnih resursa je rešen, omogućavajući programerima da se fokusiraju na podatk.

Programeri takođe mogu brzo da obezbede baze podataka po potrebi, lako klonirajući skupove podataka i konfiguracije bez potrebe za pomoći tima za IT infrastrukturu. Pružanje API-ja kao dela usluge u oblaku omogućava im da izgrade aplikacije sledeće generacije koje su potrebne preduzećima da postignu svoje strateške ciljeve.

Što brži programeri mogu da obavljaju ove relativno jednostavne administrativne zadatke, to pre mogu da isporuče ažuriranja koda i poboljšanja koja pojednostavljuju operacije zasnovane na podacima.

## Operacije sa podacima za buduće upotrebe

Odabirom *NoSQL* baze podataka *DBaaS* kao što je *MongoDB* *Atlas* proširuje se potencijal operacija. Nestruktuirano upravljane bazom podataka može se koristiti za isporuku podataka kao usluge izgradnjom operativnog sloja podataka (ODL) nad NoSQL bazom podataka.

ODL čini sve korporativne podatke dostupnim na zahtev, spremnim za izgradnju novih transformacionih aplikacija koje pomažu kompaniji da učini više sa podacima koje poseduje. ODL je važan korak ka izgradnji inteligentnih, brzih aplikacija u realnom vremenu.

## Korišćenje *DBaaS*-a

Organizacije poslovne klase koriste SQL i NoSQL baze podataka za izgradnju ogromnih skladišta podataka za pokretanje svojih operacija analize velikih podataka u realnom vremenu. Provajderi kao što su *Microsoft Azure* i *Google Cloud Platform* upravljaju infrastrukturom i platformom, omogućavajući klijentima da se usredsrede na izgradnju složenih aplikacija u realnom vremenu koje pokreću njihove operacije.

DBaaS je dobar za svaku aplikaciju kojoj je potrebna skalabilnost i fleksibilnost za njihove baze podataka, njegovo korišćenje je u rapidnom porastu.

## Izbor *DBaaS*-a

Izbor *DBaaS*-a zavisi od operativnih potreba. U mnogim slučajevima, potencijalno se može izabrati slično rešenje kao postojeće, migrirajući postojeću relacionu bazu podataka na ekvivalent u oblaku.

Alternativno, možete se izabrati za nestrukturisana NoSQL baza podataka, koja nudi maksimalnu fleksibilnost ako se podaci budu promenili u budućnosti. *MongoDB* je nerelaciona baza podataka koja omogućava da čuvanje neobrađenih, neuređenih podatke, obezbeđujući da svaki detalj ostane dostupan ako se operativne potrebe promene.

Svi tipovi baza podataka imaju svoje prednosti, ali izbor prave baze podataka treba da bude vođen slučajem upotrebe i stručnošću vašeg inženjerskog tima. Nestrukturirane baze podataka nude veću fleksibilnost i skalabilnost, ali po cenu gubitka transakcijskih garancija koje pruža strukturirana alternativa.

## Prednosti *DBaaS*-a

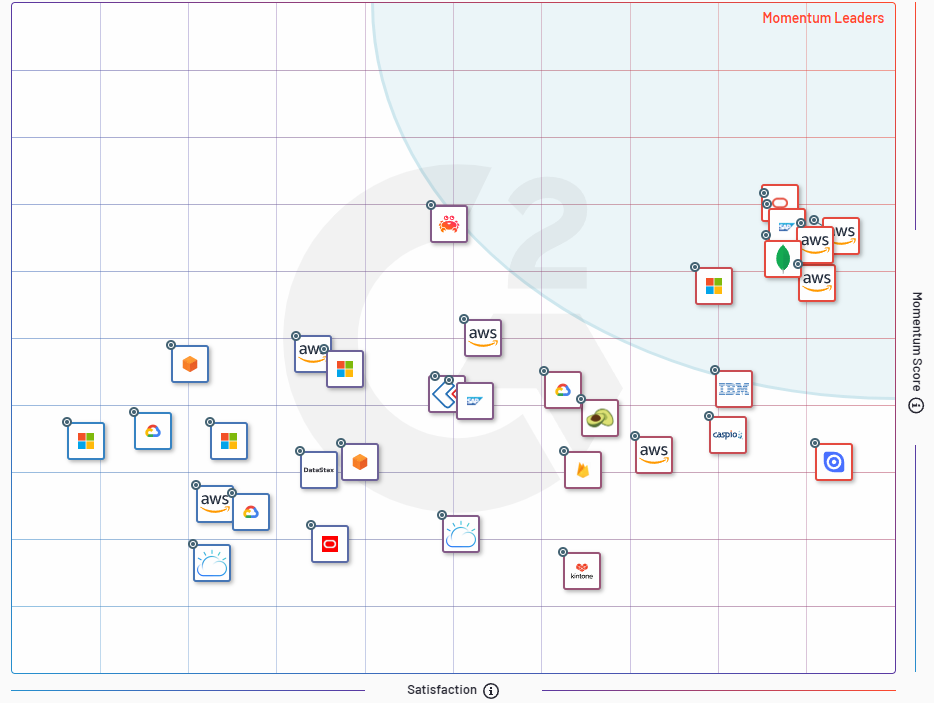
* *DBaaS* je odgovoran za dobavljača informacija za upravljanje i održavanje hardvera i koda.
* Eliminisani su veliki računi za struju, ventilaciju i hlađenje kako bi se zadržala jedinica za rad servera.
* Organizacija koja je pretplaćena na *DBaaS* ne može da angažuje programere informacija ili da sama pravi informacioni sistem.
* Ljudski resursi potrebni za upravljanje održavanjem sistema su eliminisani.
* Pošto se *DBaaS* hostuje van lokacije, organizacija nema problema sa napajanjem ili kvarom na mreži.

## Nedostaci DBaaS-a

* Tradicionalne kompanije mogu generalno imati primedbe na usluge zasnovane na oblaku.
* U slučaju značajnog kvara *DBaaS* servera ili mreže, organizacija može izgubiti svoje podatke.
* Kompanije koje su već opremljene resursima i ljudskim resursima vezanim za IT možda neće shvatiti da su *DBaaS* rešenja ekonomski održiva.
* Problemi povezani sa mrežom i konekcijom sa oblakom mogu uticati na performanse *DBaaS*-a.
* Funkcije koje se nude u okviru tipičnog *RDBMS*-a možda neće biti stalno ponuđene tokom *DBaaS* sistema.
* Upotreba *DBaaS*-a može dovesti do gubitka prihoda u alternativnim oblastima ažuriranja koda i upravljanja hardverom.

# **Primeri *DBaaS*-a**

Na narednom dijagramu su prikazani najboljih *DBaaS*-obi prema sajtu <https://www.g2.com> . G2 ocena je zasnovana na recenzijama prikupljenim od strane korisničke zajednice, kao i na podacima prikupljenim iz onlajn izvora i društvenih mreža. Jedinstveni algoritam se primenjuje na ove podatke za izračunavanje rezultata zadovoljstva i prisustva na tržištu u realnom vremenu.



*Slika 1 – Rangiranje DBaaS-a*

Deset najboljih provajdera baza podataka kao usluge (DBaaS):

* Amazon Relational Database Service (RDS)
* Amazon DynamoDB
* Oracle Database
* Amazon Athena
* SAP HANA Cloud
* IBM Db2
* MongoDB Atlas
* Azure SQL Database
* Amazon Aurora
* Ninox Database

MongoDB Atlas

asfkasfsaf

Zaključak

Kada se čuvaju podaci u *MongoDB*-u, može biti više *Mongo* servera (zvanih replike) koji drže kopiju uskladištenih podataka. Na ovaj način, čak i ako se jedan od servera pokvari, imamo rezervnu kopiju za preuzimanje podataka. Ova tehnika se koristi da bi aplikacije bile otpornije na kvarove baze podataka, pošto aplikacija može nastaviti da radi čak i ako jedan server baze podataka nije u funkciji.

Reference

[1] https://www.mongodb.com/docs/manual/core/replica-set-members/

[2] https://www.mongodb.com/docs/manual/core/replica-set-secondary/

[3] https://www.mongodb.com/docs/manual/core/read-preference/

[4] https://www.mongodb.com/docs/manual/core/replica-set-priority-0-member/

[5] https://www.mongodb.com/docs/manual/core/replica-set-hidden-member/

[6] https://www.mongodb.com/docs/manual/core/replica-set-delayed-member/

[7] https://www.mongodb.com/docs/manual/core/replica-set-arbiter/

[8] https://www.mongodb.com/docs/manual/core/replica-set-oplog/

[9] https://www.mongodb.com/docs/manual/core/replica-set-sync/

[10] https://www.mongodb.com/docs/manual/core/replica-set-architectures/

[11] https://www.mongodb.com/docs/manual/core/replica-set-high-availability/

[12] https://www.mongodb.com/docs/manual/applications/replication/

[13] https://www.bmc.com/blogs/mongodb-replication/

[14] https://www.tutorialspoint.com/mongodb/mongodb\_replication.htm

[15] https://hevodata.com/learn/mongodb-replica-set-3-easy-methods/

[16] https://www.sohamkamani.com/docker/mongo-replica-set/