

**ELEKTRONSKI FAKULTET NIŠ**

# **SEMINARSKI RAD**

**PREDMET:**

**Sistemi za upravljanje bazama podataka**

**TEMA:**

**High availability rešenja**

**Profesor:**

**Doc. Dr Aleksandar Stanimirović**

**Student:**

**Tijana Spasić 1064**

**Niš, Jun 2020**

<b>Visoka pouzdanost sistema</b>	<b>3</b>
<b>Koncepti visoke pouzdanosti</b>	<b>4</b>
<b>Microsoft SQL Server tehnike za visoku pouzdanost sistema</b>	<b>5</b>
Log shipping	5
Primer SQL Server log shipping-a	6
Prednosti i nedostaci korišćenja SQL Server log shipping-a	8
Replikacija transakcije	8
Database mirroring	10
Primer SQL Server database mirroring-a	11
Prednosti i nedostaci korišćenja database mirroring-a	12
Always on Failover Cluster	12
Primer SQL Server Always on Failover Cluster-a	14
Always on Availability Groups	15
<b>Literatura</b>	<b>17</b>

# Visoka pouzdanost sistema

Živimo u svetu zasnovanom na IT-u, u kome uspeh i rast poslovanja uglavnom zavise od raspoloživosti kritičnih sistema koji ga vode. Nije iznenađujuće da se tokom sklapanja sporazuma o nivou usluge (engl. Service Level Agreement, SLA) pominju teme koje uključuju cilj vremena oporavka (RTO) i cilj tačke oporavka (RPO), kao i da sekunda zastoja u finansijskim sistemima ili sistemima praćenja neće biti prihvaćena.

Dostupnost poslovnih aplikacija zavisi i od dostupnosti podataka koje te aplikacije obrađuju i čuvaju. Koja je potreba za aplikacijom koja je dostupna bez mogućnosti pristupa do baza podataka koje ova aplikacija koristi?

Od administratora baza podataka u nekoj međunarodnoj kompaniji ili kompaniji koja pruža finansijske usluge, biće traženo da obezbedi pouzdano rešenje, visoke dostupnosti i oporavka od katastrofe koje osigurava minimalni mogući prekid rada sistema baza podataka u slučaju bilo kakvih padova, neuspeha ili katastrofalnih situacija. *Microsoft* pruža brojne tehnologije koje se mogu lako koristiti za dizajniranje pouzdanog rešenja visoke dostupnosti i oporavka od katastrofe.

# Koncepti visoke pouzdanosti

Pre nego što bude reči o različitim tehnologijama dostupnosti i oporavka od katastrofe, potrebno je razumeti razliku između koncepata velike dostupnosti i oporavka od katastrofe. Velika dostupnost znači da će SQL Server instance ili baze podataka biti dostupne i dostižne, sa najmanje moguće zastoja, u slučaju pada ili nestanka servera. Dostupnost SQL Servera koji pruža rešenje meri se brojem 9-ki u procentu raspoloživosti koji ovo rešenje pruža i ukupnim periodom godišnjeg zastoja, prikazano u *Microsoft*-ovoj tabeli 1.1.:

Number of 9's	Availability Percentage	Total Annual Downtime
2	99%	3 days, 15 hours
3	99.9%	8 hours, 45 minutes
4	99.99%	52 minutes, 34 seconds
5	99.999%	5 minutes, 15 seconds

Tabela 1.1. Dostupnost SQL Servera

S druge strane, oporavak od katastrofe znači da, kada primarni centar podataka doživi katastrofalni događaj, poput zemljotresa, poplave ili rata, sisteme treba dovesti na mrežu i omogućiti da odatle služe korisnicima.

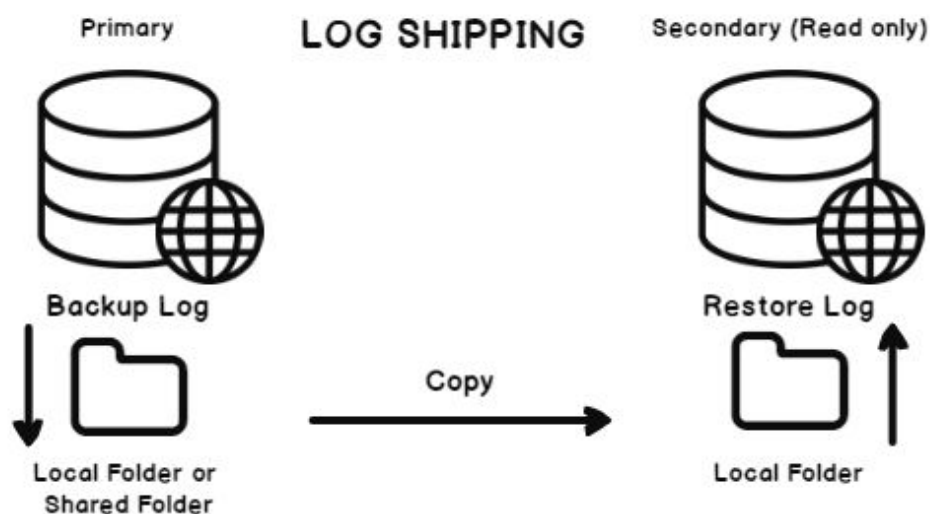
Microsoft SQL Server pruža različite tehnologije visoke dostupnosti.

# Microsoft SQL Server tehnike za visoku pouzdanost sistema

## Log shipping

SQL Server log shipping predstavlja rešenje visoke dostupnosti na nivou baze podataka koje se uklapa u manje kritične baze podataka sa fleksibilnom tačkom oporavka i vremenom oporavka. Sastoji se od jednog primarnog servera baze podataka i jednog ili više sekundarnih servera u stanju pripravnosti koji se mogu koristiti u svrhe izveštavanja.

Rad funkcije *log shipping*-a zahteva da baza podataka bude u modelu potpunog oporavka jer jako zavisi od log transakcija. Počinje kopiranjem primarne baze podataka na sekundarne servere, a zatim će sekundarna baza podataka biti sinhronizovana kumulativnim sigurnosnim kopijama log transakcija SQL Server-a koje su preuzete iz primarne baze podataka. Ovo će se izvršiti automatski korišćenjem tri SQL Server Agent posla. Prvi posao preuzeće rezervnu kopiju log transakcija iz primarne baze podataka, drugi posao će kopirati tu sigurnosnu kopiju log transakcija na sekundarne servere, a treći posao će vratiti ove sigurnosne kopije log transakcija u sekundarne baze podataka, bez izvođenja postupka oporavka u sekundarnim bazama podataka. To znači da se sekundarne baze podataka neće objavljivati na mreži. Umesto toga, biće u oporavku ili stanju pripravnosti, čekajući sledeću sigurnosnu kopiju log transakcija SQL Server-a sa primarnog servera, kao što je prikazano na slici 3.1.:



Slika 3.1. Log shipping

Neuspeh između primarnog i sekundarnog servera u *log shipping*-u, poznat kao promena uloge, trebao bi se izvršiti ručno pomoću T-SQL naredbe.

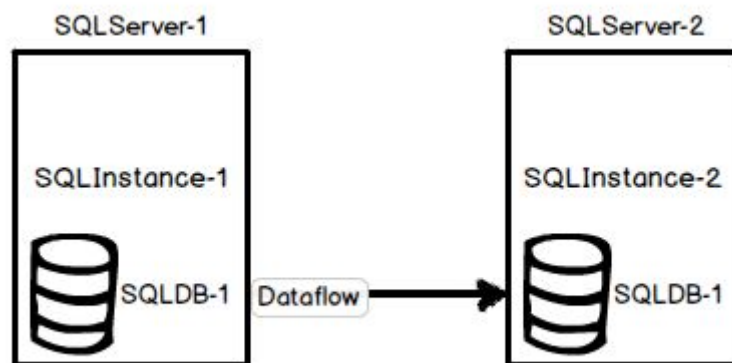
SQL Server log shipping je tehnika koja uključuje dve ili više instanci SQL Servera i kopiranje datoteke log transakcija iz jedne instance SQL Servera u drugu. Proces prenosa log transakcija i njihovo obnavljanje je automatizovan preko samog SQL Servera. Kao rezultat procesa, postoje dve kopije podataka na dve odvojene lokacije.

Log shipping sesija obuhvata sledeće korake:

- Izrada sigurnosne kopije log transakcija na primarnoj instanci SQL Servera
- Kopiranje datoteke sigurnosne kopije log transakcija kroz mrežu u jednu ili više sekundarnih instanci SQL Servera
- Vraćanje datoteke sigurnosne kopije log transakcija na sekundarnim instancama SQL Servera

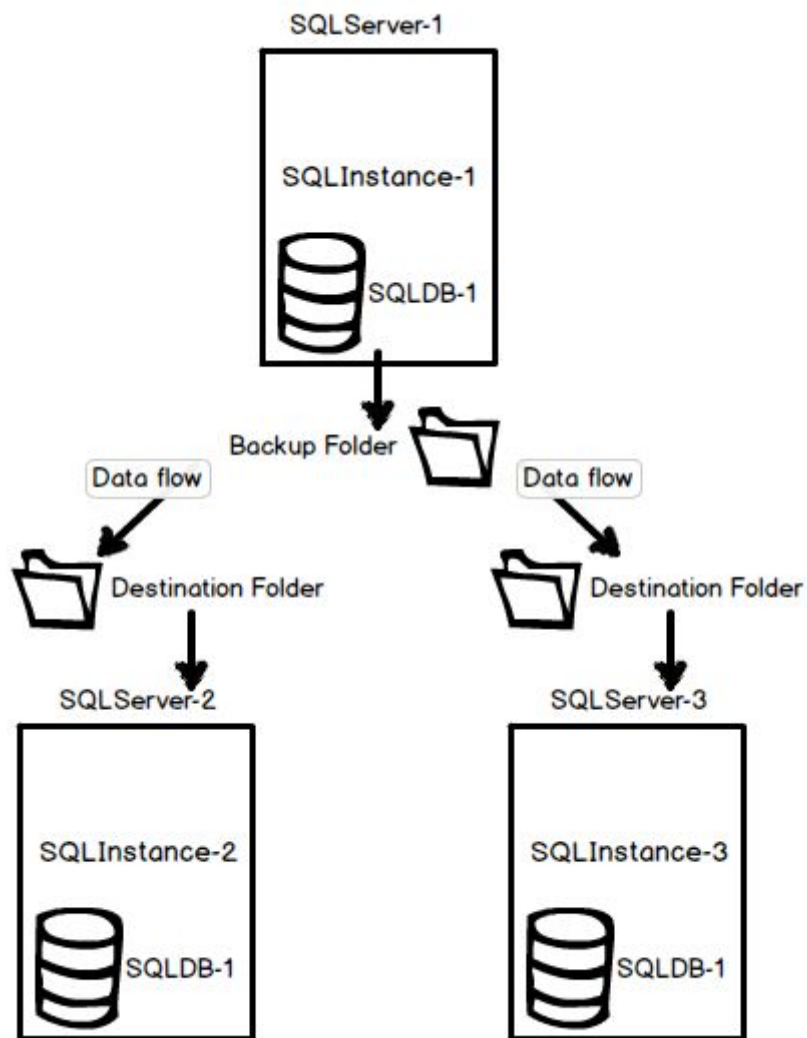
### Primer SQL Server log shipping-a

Jedan od uobičajenih log shipping scenarija, prikazano na slici 3.2, je okruženje sa dva servera (SQLServer-1 - primarni i SQLServer-2 - sekundarni), dve instance SQL Servera (SQLInstance-1 i SQLInstance-2) i jednom bazom podataka SQL Server pod nazivom SQLDB-1 sa pokrenutim log shipping-om na njemu.



Slika 3.2. Primer 1 - SQL Server log shipping-a sa dva servera

Druga uobičajena konfiguracija, prikazano na slici 3.3, je okruženje sa tri (ili više) servera (SQLServer-1 - primarni, SQLServer-2 - sekundarni i SQLServer-3 - sekundarni), tri instance SQL Servera (SQLInstance-1, SQLInstance-2 i SQLInstance-3) i jednom bazom podataka SQL Servera pod nazivom SQLDB-1 sa pokrenutim log shipping-om na njemu.



Slika 3.3. Primer 2 - SQL Server log shipping-a sa tri ili više servera

Postoje dva režima rada i oni su povezani sa stanjem u kojem će biti sekundarna baza podataka SQL Servera:

- Stanje pripravnosti - baza podataka dostupna je za upite i korisnici joj mogu pristupiti, ali u režimu samo za čitanje
  - Baza podataka nije dostupna samo dok traje postupak vraćanja
    - Korisnici mogu biti prisiljeni da prekinu vezu kada započne posao vraćanja
    - Posao vraćanja može da se odloži sve dok se svi korisnici ne prekinu
- Režim obnavljanja - baza podataka nije dostupna

## Prednosti i nedostaci korišćenja SQL Server log shipping-a

SQL Server log shipping uglavnom se koristi kao rešenje za oporavak od katastrofe. Korišćenje log shipping-a putem SQL Server-a ima višestruke prednosti: pouzdano je i detaljno testirano, relativno je jednostavno podešavanje i održavanje, postoji mogućnost zabrane prolaska između SQL servera, podaci se mogu kopirati na više lokacija itd.

Log shipping se može kombinovati sa drugim opcijama za oporavak od katastrofe, kao što su AlwaysOn Availability Groups, mirroring i replikacija baze podataka. Takođe, SQL Server log shipping ima niske troškove u ljudskim i serverskim resursima.

Glavni nedostaci u SQL Server log shipping tehnici su: potreba da se upravlja svim bazama podataka odvojeno, ne postoji mogućnost automatskog prelaska, a sekundarna baza podataka nije u potpunosti čitljiva dok traje proces vraćanja.

## Replikacija transakcije

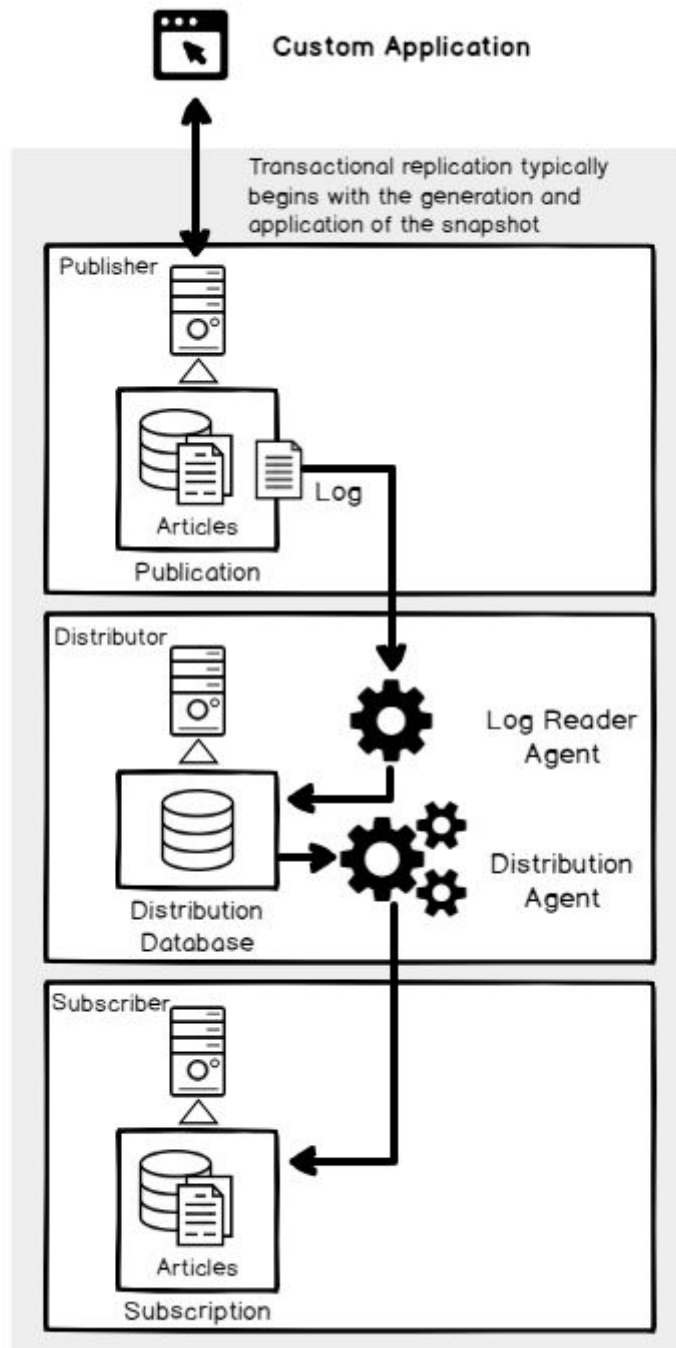
Transakcijska replikacija SQL Server-a koja se odvija u realnom vremenu, na nivou baze podataka, je rešenje visoke dostupnosti, koje se sastoji od jednog primarnog servera, poznatog kao *publisher*, koji sve tabele baze podataka ili odabrane tabele, poznate kao artikli, distribuira na jedan ili više sekundarnih servera, poznatih kao *subscriber*-i, koji se mogu koristiti i za potrebe izveštavanja.

Kao što ime kaže, replikacija transakcije zavisi od procesa sinhronizacije sa SQL Server transaction log-ovima.

Radi uz pomoć tri agenta:

- Snapshot agent SQL Servera koji priprema početni snapshot fajl koji sadrži objekte baze podataka koje treba replicirati, može se predstaviti kao potpuna sigurnosna kopija
- Agent za distribuciju odgovoran je za kopiranje početnog snapshot fajla i nagomilanih logova pretplatnicima
- Log reader agent odgovoran je za nadgledanje log transakcija SQL Server-a u bazi izdavača i kopiranje ovih transakcija iz datoteke log transakcija te baze podataka u distribucionu bazu podataka, da bi ih distributivni agent kasnije kopirao na pretplatnike, kao što je prikazano na slici 3.4:





Slika 3.4. Primer funkcionisanja transakcijske replikacije

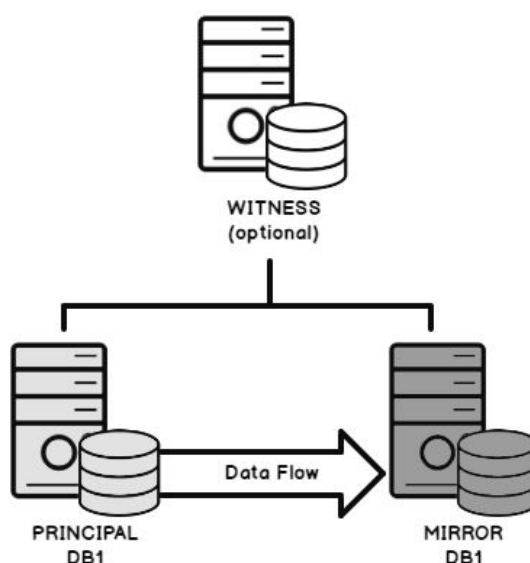
## Database mirroring

Database mirroring SQL Server-a je rešenje visoke dostupnosti na nivou baze podataka koje se može konfigurirati nad bazama podataka sa potpunim modelom oporavka. Sastoji se od najmanje dva servera; primarni SQL Server, poznat kao **Principal** server i sekundarni server, poznat kao **Mirror** server, i opciono treći server, poznat i kao **Witness** server. Witness server će pratiti vezu između ova dva servera i njihovu dostupnost i izvršiti automatsko dupliranje fajla, ili promenu uloge, između ova dva servera.

Database mirroring SQL Server-a radi u dva sinhronizaciona moda:

- Režim visoke sigurnosti, poznat i kao sinhroni režim, u kome će se transakcija izvršiti na glavnoj bazi podataka nakon što je izvrši i upisati u datoteku log transakcija u bazu podataka ogledala (engl. database mirror), što povećava mogućnost kašnjenja transakcija
- U režimu sinhronizacije visokih performansi, takođe poznat kao asinhroni režim, transakcija će se izvršiti na glavnoj bazi podataka bez čekanja da se izvrši na mirror serveru, smanjujući mogućnost kašnjenja transakcija, ali povećavajući mogućnost gubitka podataka

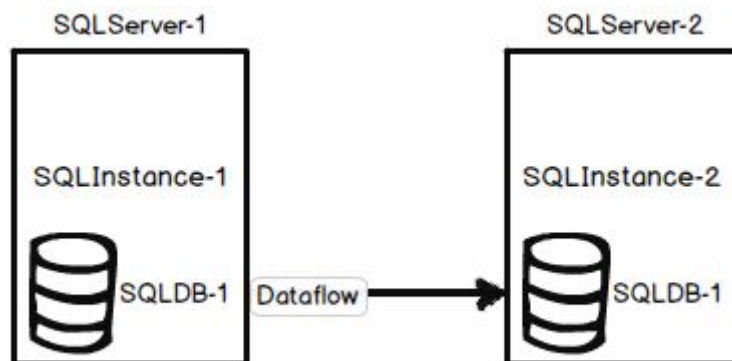
Database mirroring pokreće se obnavljanjem potpune sigurnosne kopije i sigurnosne kopije SQL Server Transaction Log-a iz glavne baze podataka na mirror serveru, bez dovođenja mirror baze podataka u mrežu. Nakon konfigurisanja mirror baze podataka, ona će biti sinhronizovana slanjem aktivnih zapisa log transakcija u mirror bazu podataka i ponavljanjem svih ovih operacija nad mirror bazom, kao što je prikazano na slici 3.5:



Slika 3.5. Database mirroring

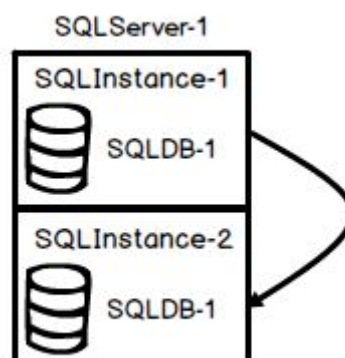
## Primer SQL Server database mirroring-a

Jedna od uobičajenih konfiguracija ogledala je okruženje sa dva SQL servera (SQLServer-1 i SQLServer-2), dve instance (SQLInstance-1 i SQLInstance-2) i jednom mirror bazom podataka nazvanom SQLDB-1, kao što je to prikazano na slici 3.6:



Slika 3.6. Primer 1 - Database mirroring sa dve mašine SQL Servera

Druga uobičajena konfiguracija, kao što je to prikazano na slici 3.7, je okruženje sa jednom mašinom SQL Server, dve instance SQL Server i jednom mirror bazom podataka nazvanom SQLDB-1. Ovo rešenje ima veliku manu, jer ako SQLServer-1 padne, obe instance neće biti dostupne.



Slika 3.7. Primer 2 - Database mirroring sa jednom mašinom SQL Servera

## Prednosti i nedostaci korišćenja database mirroring-a

Korišćenje mirror baze podataka SQL Servera ima višestruke prednosti: ugrađena funkcija SQL Servera, relativno jednostavna za podešavanje, može pružiti automatsko prebacivanje u režimu visoke sigurnosti, itd. Mirror baza podataka može se kombinovati sa drugim opcijama za oporavak od katastrofe, kao što su klasteriranje, slanje logova i replikacija.

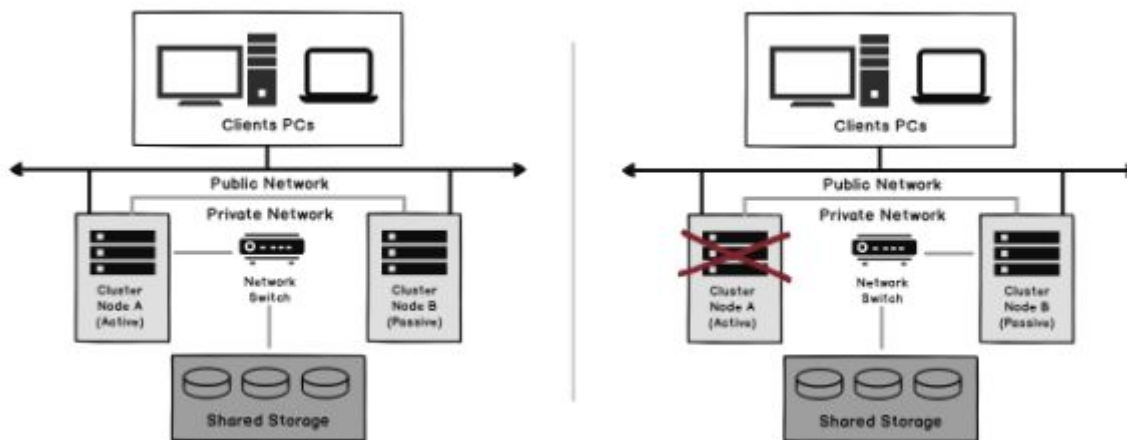
Database mirroring biće uklonjen iz SQL servera u budućim verzijama u korist AlwaysOn Availability Groups. Takođe, database mirroring je rešenje samo za bazu podataka, što znači da se prijava i poslovi sa glavnog SQL servera moraju ručno ponovo stvoriti na mirror-u. Postoji i mogućnost kašnjenja, koja se može smanjiti samo boljim hardverom.

## Always on Failover Cluster

SQL Server Always on Failover Cluster je rešenje visoke dostupnosti na nivou instance koje se gradi preko *Windows Server Failover Clustering* funkcije. Sastoji se od velikog broja servera, poznatih kao čvorovi klastera, koji imaju iste hardverske i softverske komponente, kako bi se omogućila velika dostupnost instanci klastera putem te redundantnosti .

Kada je SQL Server Failover Cluster konfiguriran i pokrenut, usluge SQL Servera i grupe resursa, uključujući deljenu memoriju, naziv mreže i virtuelne IP adrese, u datom trenutku mogu biti vlasnici samo jednog od čvorova klastera.

Ako se bilo kakav kvar, poput kvara operativnog sistema, hardvera ili usluge, dogodio na aktivnom čvoru koji je vlasnik grupa resursa i na mreži ima SQL Server uslugu ili će se jednostavno izvršiti planirano ponovno pokretanje ili nadogradnja za taj aktivni čvor, vlasništvo nad grupom resursa biće u potpunosti preseljeno u drugi čvor klastera, gde će se instanca SQL Servera prebaciti van mreže na prethodni aktivni čvor, a zatim preneti na mrežu na novom čvoru koji je vlasnik grupe resursa, kao što je prikazano na slici 3.8:



Slika 3.8. SQL Server Always on Failover Cluster

U SQL Server Always on Failover Cluster-u, ne koristi se log transakcija SQL Servera, jer se konfiguracija vrši na nivou instancije SQL Servera, bez potrebe za sinhronizacijom izmena između različitih čvorova. Drugim rečima, baza podataka SQL Servera će biti smeštena i na mreži na jednom čvoru u određeno vreme, bez replika za tu bazu podataka na ostalim čvorovima klastera.

SQL Server Always on Failover Cluster sastoji se dakle od grupe servera koji pokreću aplikacije za klaster na poseban način, tako da umanjuju zastoj. Prelazak na neuspeh (engl. failover) je proces koji se dogodi ako se jedan čvor sruši ili postane nedostupan, a drugi preuzme i automatski pokrene aplikaciju bez ljudske intervencije.

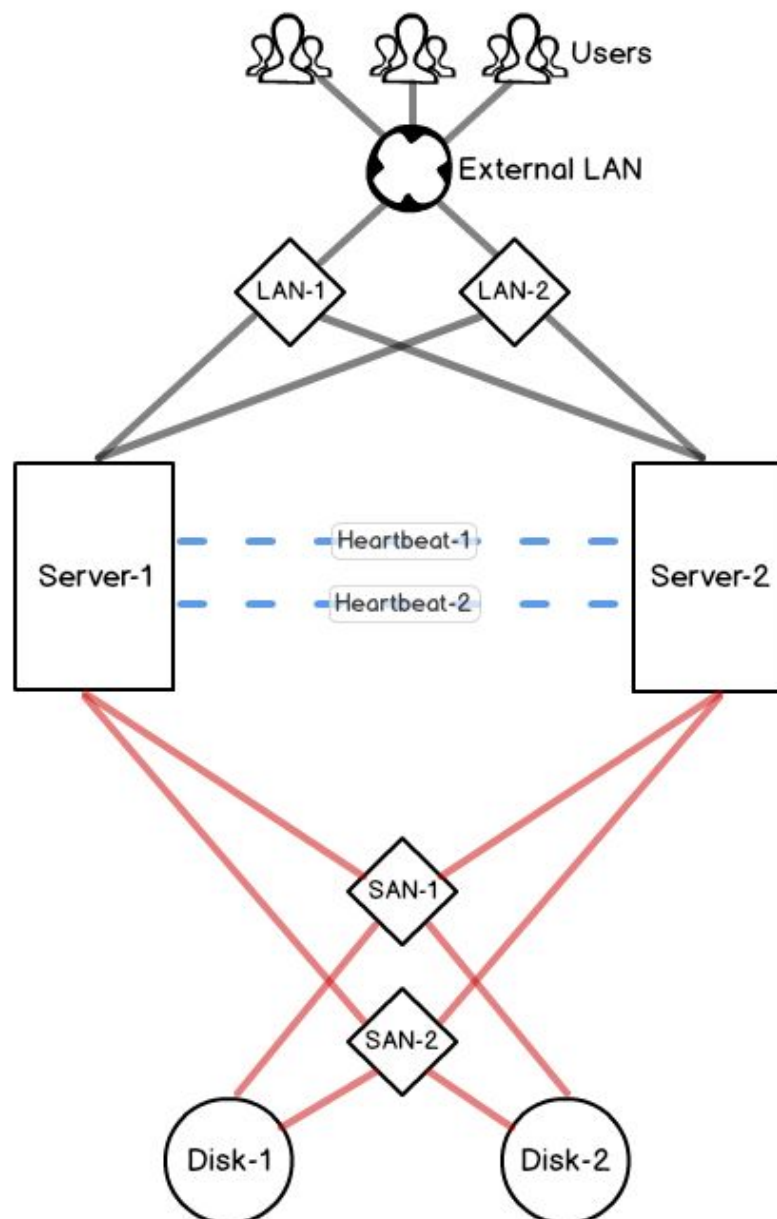
Glavni koncept koji se nalazi iza *failover* klastera je uklanjanjanje jedne pristupne tačke uključivanjem više mrežnih veza i deljenog prostora za podatke spojenih preko SAN-a (Storage area network) ili NAS-a (Network storage storage).

Svaki čvor klastera prati se sve vreme putem privatne mrežne veze zvane *heartbeat*. Sistem mora biti u stanju da prevaziđe situaciju koja se zove „*split-brain*“ koja se dogodi ako sve veze otkucaja srca padnu istovremeno. Zatim svi drugi čvorovi mogu zaključiti da je jedan čvor ugašen i pokušati ponovo pokrenuti aplikaciju na sebi. Klaster prelaska koristi se na bazi kvoruma za nadgledanje ukupnog zdravlja klastera i maksimiziranje tolerancije na greške na nivou čvora.

## Primer SQL Server Always on Failover Cluster-a

Klaster pod nazivom CLUSTER-01 sadrži dva server-čvora, nazvane CLUSTER-01-SRV-01, CLUSTER-01-SRV-02. Postoji jedna instanca SQL Servera pod nazivom SQL-INST-01. Takođe, sva tri servera dele zajedničku memoriju.

Kada se server CLUSTER-01-SRV-01 sruši, failover klaster servis u CLUSTER-01 prepoznaje situaciju kroz otkucaje srca i automatski pokreće instancu SQL Servera SQL-INST-01 na CLUSTER-01-SRV-02 serveru, slika 3.9.



Slika 3.9. Primer SQL Server Always on Failover Cluster-a

U SQL Server failover klasteru podaci moraju biti u zajedničkoj memoriji. Klaster može premestiti instancu SQL Servera ako jedan čvor ima problema jer se dele svi podaci. Ovo rešenje može garantovati veće vreme trajanja i višak zaliha. Budući da postoji samo jedan prostor za skladištenje, potrebni su redovni zahtevi za održavanje SQL Servera. Takođe, ako zajednička memorija nije suvišna, nakon neuspeha u skladištu, baza podataka SQL Servera neće biti dostupna. Za zauzeta okruženja SQL Servera, gde se prekid meri u sekundi, treba razmotriti vreme „pada“ jer promena između čvorova nije trenutna.

## Always on Availability Groups

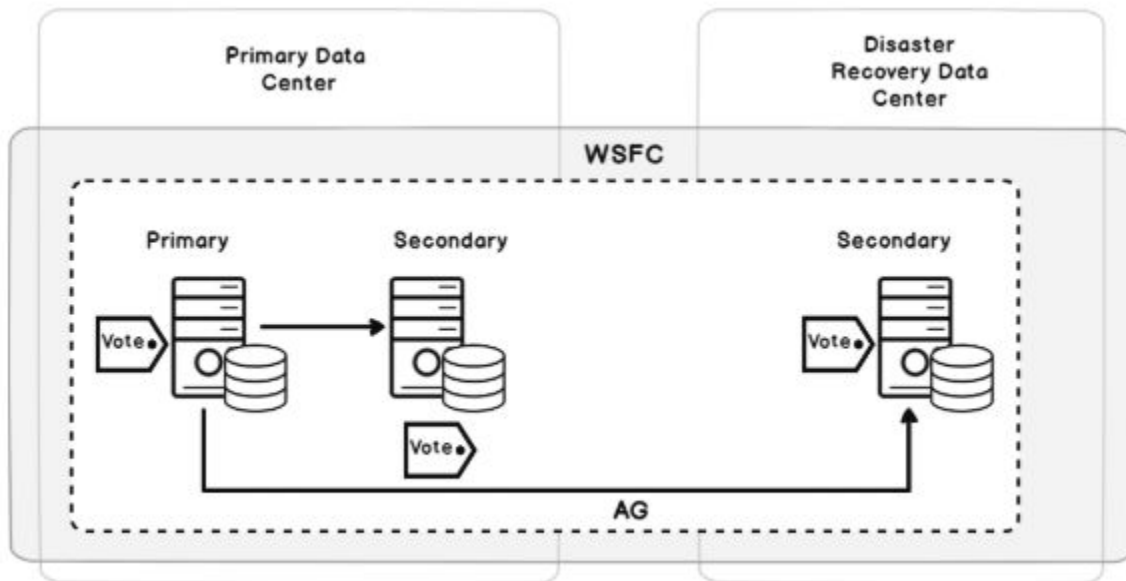
SQL Server Always on Availability Group podrška, koja je prvi put predstavljena u SQL Serveru 2012, je rešenje visoke dostupnosti na nivou baze podataka koje je izgrađeno preko *Windows Server Failover Clustering* funkcije. Sastoji se od jednog primarnog servera, poznatog kao primarna replika, u kojem će baze podataka biti dostupne za serviranje veza za čitanje i upis i do osam sekundarnih servera, poznatih kao sekundarne replike, koji se mogu koristiti za serviranje veza samo za čitanje za potrebe izveštavanja.

Grupa dostupnosti je skup korisničkih baza podataka koje se međusobno preklapaju između replika dostupnosti. Drugim rečima, pitanje nivoa baze podataka, kao što je oštećenje baze podataka ili gubitak podataka, neće primorati da se grupa dostupnosti prebaci na drugu repliku, pri čemu neuspeh baze podataka neće uticati na ostale baze podataka u istoj grupi dostupnosti.

Da bi se korisnici baze podataka lakše povezali sa grupom dostupnosti bez potrebe da znaju koja instanca SQL Servera je domaćin primarnoj replici, može se stvoriti *Listener* grupa dostupnosti. Ova grupa je *virtual network name*, koje se sastoji od jedinstvenog DNS imena, jedne ili više virtuelnih IP adresa i TCP broja porta koji omogućava direktnu vezu sa odgovarajućom kopijom grupe dostupnosti, usmeravanjem čitanja i pisanja radno opterećenje do primarne replike i usmeravanje radnog opterećenja samo za čitanje na sekundarnu repliku, ako je lista usmeravanja samo za čitanje konfigurirana na toj grupi dostupnosti.

Sinhronizacija podataka između primarne replike i sekundarne replike odvija se na nivou baze podataka i jako zavisi od SQL Server Transaction Log-a te baze podataka. Primarna replika šalje zapise log transakcija iz svake baze podataka koja učestvuje u grupi raspoloživosti u sve sekundarne replike. Na drugoj strani, sekundarne replike će te zapise transakcija upisati u datoteku log transakcija baze podataka radi keširanja, a zatim će ih ponovo zapisati u odgovarajuću sekundarnu bazu podataka. Ako je sekundarna replika konfigurisana za režim asinhronog zauzimanja, primarna replika neće čekati da sekundarna replika upiše dolazne zapise dnevnika transakcija na disk. Uzimajući u obzir da se promene

ili promene na nivou servera koje nisu zapisane u datoteku log transakcija SQL Server-a, kao što su prijave, povezani serveri i poslovi SQL agenta neće automatski sinhronizovati između replika grupe dostupnosti i zahtevati da sinhronizujete ga ručno u sekundarnim replikama.



Slika 3.10. Always on availability groups



# Literatura

[1] Matt Lee, "The ultimate guide to High Availability methods for Microsoft SQL Server"

Web-sajt

<https://blog.papercut.com/the-ultimate-guide-to-high-availability-methods-for-microsoft-sql-server/>

(pristupljeno juna 2020)

[2] "SQL Server High Availability Options"

Web-sajt <https://www.mssqltips.com/sqlservertip/2482/sql-server-high-availability-options/>

(pristupljeno juna2020)

[3] Ivan Stanković, "What is SQL Server log shipping?"

Web-sajt <https://www.sqlshack.com/sql-server-log-shipping/> (pristupljeno juna2020)

[4] Ivan Stanković, "SQL Server transactional replication"

Web-sajt

<https://www.sqlshack.com/sql-server-transactional-replication-how-to-reinitialize-a-subscription-using-a-sql-server-database-backup/> (pristupljeno juna2020)

[5] Ivan Stanković, "What is SQL Server database mirroring?"

Web-sajt <https://www.sqlshack.com/sql-server-database-mirroring/> (pristupljeno juna2020)

[6] Ivan Stanković, "AlwaysOn Availability Groups"

Web-sajt

<https://www.sqlshack.com/alwayson-availability-groups-curiosities-to-make-your-job-easier-part-1/> (pristupljeno juna2020)