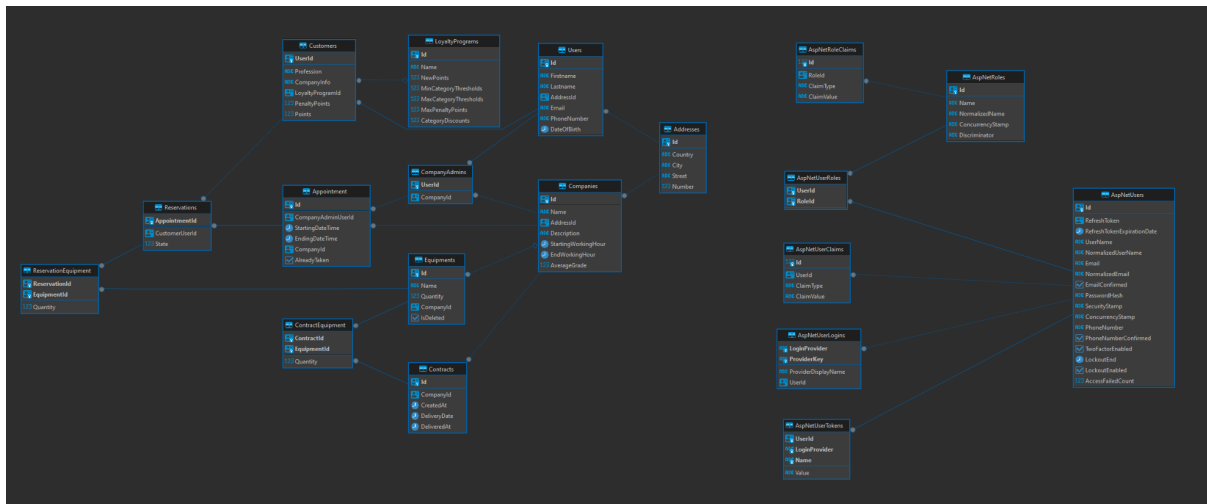


Dizajn šeme baze podataka



Preflog za particionisanje podataka

Vertikalno Particionisanje za Optimizaciju Pristupa Podacima: Da bi se olakšao i ubrzao pristup podacima, predlaže se vertikalno particionisanje tabele user tako da se osnovni podaci korisnika (kao što su korisničko ime, lozinka i penali) odvoje od ličnih podataka koji se ređe koriste i menjaju. Ovo omogućava efikasniji pristup i upravljanje često korišćenim podacima, dok se za ređe korišćene podatke može primeniti drugačiji nivo pristupa i sigurnosti.

Modularno Grupisanje Entiteta: Grupisanje entiteta po funkcionalnosti u modularnom pristupu omogućava da se smanji međuzavisnost podataka i pojednostavi upravljanje sistemom. Svaki modul, poput rezervacija, korisnika ili kompanije, može imati svoju izdvojenu bazu podataka ili skup tabela unutar jedne baze, čime se olakšava skaliranje i održavanje sistema.

Horizontalno Partitionisanje za Skalabilnost: Podaci kao što su rezervacije i termini mogu se horizontalno particionisati po vremenskim intervalima, geografskoj lokaciji ili drugim relevantnim kriterijumima. Ovo horizontalno particionisanje omogućava distribuciju podataka preko više servera ili instanci, čime se poboljšavaju performanse upita i skalabilnost sistema.

Upotreba Šema Podataka za Fleksibilnost: Umesto kreiranja novih baza podataka, može se razmotriti upotreba različitih šema unutar iste baze podataka za organizaciju particionisanih podataka. Šeme omogućavaju logičko grupisanje i izolaciju podataka unutar iste baze, čime se olakšava upravljanje i održavanje bez potrebe za fizičkom separacijom na nivou baza podataka.

Predlog strategije za replikaciju baze i obezbeđivanje otpornosti na greške

Azure Database for PostgreSQL - Fleksibilno Skaliranje

Azure Database for PostgreSQL je usluga koja pruža automatsko skaliranje, visoku dostupnost, i zaštitu podataka.

Visoka Dostupnost: Azure Database for PostgreSQL nudi opcije za visoku dostupnost koje automatski replikuju podatke na više dostupnih zona kako bi se obezbedila otpornost na regionalne prekide.

Geo-Redundantna Replikacija: Za dodatnu otpornost na greške, može se konfigurisati geo-redundantna replikacija koja sinhronizuje podatke u realnom vremenu sa sekundarnom lokacijom na Azure-u. Ovo osigurava da su podaci dostupni čak i u slučaju geografskog incidenta.

Konfiguracija Master-Slave Replikacije

Za lokalni setup i inicijalno postavljanje na Azure, može se koristiti Master-Slave replikacija koja omogućava da jedan primarni (master) server obrađuje upise, dok jedan ili više sekundarnih (slave) servera služe za čitanje i kao backup za visoku dostupnost:

Konfigurisanje Streaming Replikacije: PostgreSQL podržava streaming replikaciju koja omogućava da sekundarni serveri budu ažurirani u skoro realnom vremenu. Ovo minimizira rizik od gubitka podataka u slučaju greške na primarnom serveru.

Automatski Failover: Pgpool-II ili Patroni su mehanizmi za automatsko upravljanje failover procesom, osiguravajući da se u slučaju pada primarnog servera, jedan od sekundarnih servera može automatski promovisati u novi primarni server.

Point-in-Time Recovery (PITR)

PITR omogućava vraćanje baze podataka na tačno određeni trenutak u prošlosti, koristeći logove transakcija. Ova funkcionalnost je izuzetno korisna za oporavak od grešaka kao što su nenamerni brisani podaci ili oštećenja baze

Azure Backup Integracija: Azure nudi integraciju sa Azure Backup servisom koji može automatizovati proces backup-a.

Predlog strategije za keširanje podataka

Integracija Redis-a za Praćenje Sesija

Distribuirani Redis Keš: Korišćenje Redis-a kao distribuiranog keša omogućava skalabilno i brzo praćenje sesija ulogovanih korisnika.

WebSocket Hub Integracija: Integracijom Redis-a sa WebSocket hub-om u našem slučaju zbog jednostavnosti rešeno unutar memorije servera, možete efikasno upravljati i distribuirati poruke korisnicima u realnom vremenu. Korišćenjem Redis Pub/Sub funkcionalnosti, omogućava se da se poruke efikasno distribuiraju ciljanim sesijama.

Optimizacija i Skaliranje

Skaliranje Redis-a: Redis klaster omogućava horizontalno skaliranje keša. Redis klaster omogućava distribuciju podataka preko više čvorova, čime se povećava dostupnost i otpornost sistema na greške, kao i kapacitet keša.

Bezbednost

Bezbednost prilikom upita: S obzirom na osetljivost podataka sesija, potrebno je implementirati šifrovanje podataka u tranzitu npr. TLS autentifikaciju.

Okvirna procena za hardverske resurse potrebne za skladištenje svih podataka u narednih 5 godina

Korisnički profil: Pretpostavimo da svaki korisnički profil zauzima oko 2 KB podataka.

Profil kompanije: Svaki profil kompanije može biti veći zbog dodatnih informacija, recimo 5 KB.

Rezervacije i termini: Svaka rezervacija sa terminima i detaljima može zauzimati oko 1 KB.

Dodatni entiteti (izveštaji, žalbe, itd.): Pretpostavimo dodatnih 2 KB po rezervaciji za ove entitete.

Pored toga, uključićemo i faktore kao što su broj registrovanih korisnika i kompanija, kako bismo mogli adekvatno proceniti ukupnu potrebu za skladištenjem.

Pretpostavimo sledeće:

Ukupan broj korisnika aplikacije: 100 miliona.

Ukupan broj kompanija: Pretpostavimo da postoji 10.000 kompanija.

Broj rezervacija na mesečnom nivou: 500.000.

Ukupna godišnja potreba za skladištenjem (bez replikacije i backup-a): 207.95 GB

Ukupna potrebna količina za skladištenje za 5 godina (sa replikacijom i backup-om): 3119.23 GB (približno 3.12 TB)

Ukupni hardverski resursi potrebni za skladištenje (uključujući dodatni prostor za logove i sistemske podatke): 3743.08 GB (približno 3.74 TB)

Parameter	Vrednost
Ukupan broj korisnika	100,000,000
Ukupan broj kompanija	10,000
Broj rezervacija godišnje	6,000,000
Veličina korisničkog profila (KB)	2
Veličina kompanijskog profila (KB)	5
Veličina rezervacije (KB)	1
Dodatni entiteti po rezervaciji (KB)	2
Ukupna veličina korisnika (GB)	190.734863
Ukupna veličina kompanija (GB)	0.04768372
Ukupna veličina rezervacija (GB)	17.16614
Ukupna godišnja potreba (GB)	207.94868469
Potrebna količina za 5 godina (GB)	3119.23027039
Ukupni hardverski resursi (GB)	3743.07632446

Predlog strategije za postavljenja load balansera

Postavljanjem nginx-ovog load balansera, koji koristi round-robin metodu distribucije zahteva, omogućavamo ravnomerno raspoređivanje zahteva na servere.

Postoji funkcija kao što je težinski round robin, gde se zahtevi raspoređuju na servere ne samo redom, već uzimajući u obzir i težinu opterećenje svakog servera. Takođe, možemo definisati da sesije korisnika ostanu na istom serveru koristeći sticky sessions, što je posebno korisno za aplikacije koje zahtevaju očuvanje sesije između zahteva.

Pored toga, implementacija load balansera u cloud okruženju donosi dodatne prednosti. Cloud platforme nude mogućnost automatskog skaliranja, gde se mogu pokretati nove instance servera kada je to potrebno, na primer, u slučaju povećanja saobraćaja.

Uz cloud-based load balancer, takođe imamo korist od dodatnih sigurnosnih funkcija koje cloud platforme nude, kao što su zaštita od DDoS napada itd.

Predlog koje operacije treba nadgledati u cilju poboljšanja sistema

Performanse upita baze podataka: Nadgledanje vremena izvršavanja upita može ukazati na potrebu za optimizacijom.

Performanse API-ja: Praćenje vremena odziva API-ja za ključne operacije, kao što su pretraga kompanija, pretraga opreme, i proces rezervacije, omogućava identifikaciju bottleneck-a u sistemu.

Učestalost i vrste grešaka: Analiza logova grešaka može otkriti ponavljajuće probleme, loše performanse određenih komponenti ili sigurnosne propuste koje je potrebno adresirati.

Korišćenje resursa: Nadgledanje upotrebe CPU-a, memorije, i mrežnog saobraćaja od strane servera i baza podataka može pomoći u identifikaciji potrebe za skaliranjem ili optimizacijom.

Sigurnosni incidenti: Praćenje pokušaja neautorizovanog pristupa, anomalija u mrežnom saobraćaju, kao i svih sigurnosnih incidenata je ključno za očuvanje integriteta i pouzdanosti sistema.

Praćenje performansi i dostupnosti sistema: Nadgledanje vremena odziva za ključne operacije kao što su logovanje, rezervacija, i procesi registracije, kao i praćenje iskorišćenja procesora i memorije, omogućava nam da identifikujemo uska grla i optimizujemo resurse sistema za periodi visoke aktivnosti.

Analiza aktivnosti korisnika: Praćenjem uobičajenih aktivnosti, pretraga, grešaka korisnika, i najčešćih operacija, dobijamo uvid u ponašanje korisnika, što nam omogućava da usmerimo razvoj aplikacije ka funkcionalnostima koje korisnici najviše koriste i cene.

Za implementaciju ovog plana koristićemo kombinaciju open source alata za monitoring i analizu, kao što su Elasticsearch, Logstash, Kibana, ...

Kompletan crtež dizajna predložene arhitekture

