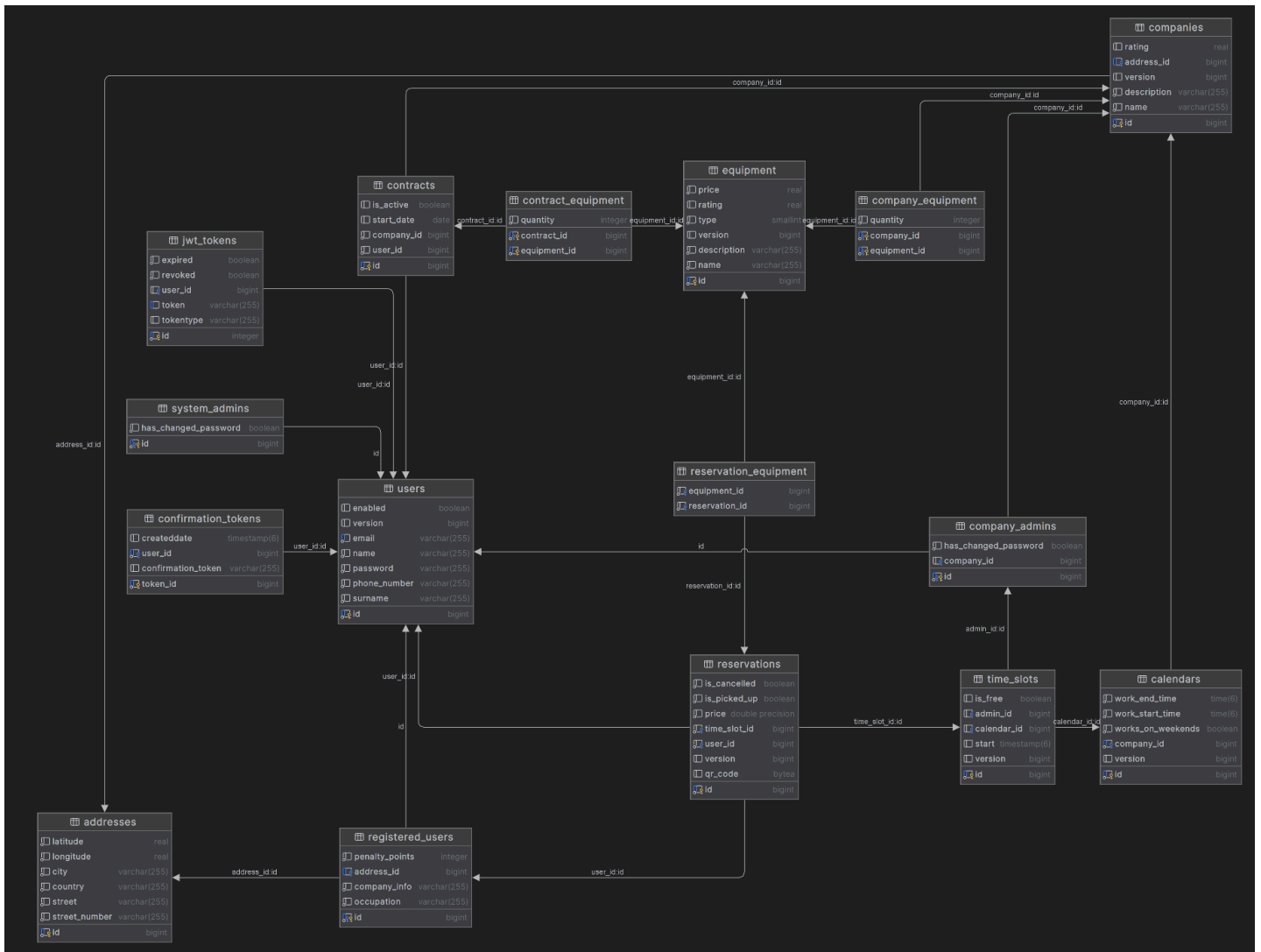


SKALABILNOST

Dizajn šeme baze podataka



Strategija za particionisanje podataka

Sa porastom broja korisnika i značajnim obimom podataka, neophodno je razviti strategiju particionisanja koja će omogućiti efikasno upravljanje velikim količinama podataka. Da bismo efikasno rešili ove izazove, doneli smo odluku o korišćenju Citus-a, ekstenzije koja omogućava distribuiranost za PostgreSQL. Citus nam pruža mogućnost da postignemo horizontalno skaliranje, fokusirajući se naročito na geografsko deljenje podataka. Implementiraćemo particionisanje podataka na osnovu geografskih lokacija. Ova tehnika omogućava smeštanje podataka o kompanijama i korisnicima iz iste regije na isti server, pružajući brži pristup podacima i poboljšavajući performanse operacija. Kompanije koje deluju u određenoj regiji sada imaju podatke smeštene na lokalnim serverima, čime se znatno ubrzava pristup podacima i poboljšavaju operativne performanse. Geografsko deljenje podataka pojednostavljuje održavanje sistema, omogućava dodavanje novih servera specifičnih za regije sa povećanim zahtevima, čime se postiže horizontalna skalabilnost. U sledećem koraku, planiramo implementaciju particionisanja koristeći ugrađeni mehanizam u PostgreSQL, konkretno putem klauzule `Partition By Range`. Ova tehnika će biti primenjena na tabelama "timeslots" i "reservations", gde ćemo organizovati podatke po datumu. Ova tehnika omogućava direktni pristup podacima za trenutni vremenski period, čime se značajno olakšava rad sa rezervacijama. Takođe, particionisanje može pomoći u optimizaciji upita koji se odnose na određene vremenske intervale, što rezultuje bržim izvršavanjem upita. Implementacijom particionisanja po datumu, smanjujemo potrebu za pretraživanjem celokupne tabele, fokusirajući se samo na određene delove podataka koji su relevantni za trenutni vremenski period. Ovo značajno unapređuje performanse sistema, posebno kada se uzme u obzir procenjenih 500.000 rezervacija mesečno. Kroz ovaj pristup, rešavamo potencijalne probleme sa efikasnošću i dugim vremenom odziva upita, čineći sistem skalabilnim i optimizovanim za rad sa rezervacijama.

Strategija za replikaciju baze i obezbeđivanje otpornosti na greške

Korišćenjem `repmgr`-a, možemo unaprediti otpornost na greške i pouzdano upravljati replikacijom podataka unutar Citus klastera. Repmgr nam omogućava automatsko otkrivanje grešaka i preusmeravanje zahteva u slučaju problema sa pojedinim čvorom. Pored toga, `repmgr` olakšava upravljanje master-slave replikacijom, automatski promovišući novi master čvor ukoliko dođe do otkaza trenutnog. Ovaj pristup osigurava neprekidnu dostupnost sistema, minimizuje vreme nedostupnosti i obezbeđuje da se podaci kontinuirano sinhronizuju između čvorova u klasteru. Kako bismo dodatno poboljšali bezbednost podataka, potrebno je implementirati redovne i automatske backup procedure. Ovo se može postići

podešavanjem **PostgreSQL** baze podataka da automatski pravi sigurnosne kopije podataka u određenim vremenskim intervalima. **Repmgr** takođe može biti konfigurisan da prati ove backup-ove i obezbedi jednostavno obnavljanje podataka u slučaju da je to potrebno.

Strategija za keširanje podataka koristeći AWS service

Implementacija efikasne strategije keširanja podataka igra ključnu ulogu u unapređenju perfansi aplikacije pružajući brz pristup često korišćenim informacijama i time umanjujući vreme odziva i opterećenje na backend servise, pa je tako korišćenje **AWS** servisa za keširanje idealno rešenje.

Za keširanje na nivou podataka može se koristiti **Amazon ElastiCache**, gde njegova integracija u našu strategiju može značajno unaprediti performanse i efikasnost aplikacije. **ElastiCache** nam pruža sposobnost da efikasno implementiramo keširanje podataka bez potrebe za kompleksnim održavanjem infrastrukture. Korišćenjem **ElastiCache**-a s **Redis**-om, postavljamo temelj za brzo i pouzdano keširanje podataka, smanjujući tako opterećenje našeg backend sistema. Osim toga, **ElastiCache** omogućava na stotine miliona operacija u sekundi sa vremenom odziva u mikrosekundama, smanjuje opterećenje na backend bazu podataka, poboljšavajući brzinu odgovora na upite, a integracija sa **Amazon CloudFront**-om omogućiće brzu isporuku keširanih kopija podataka širom sveta. Za distribuirano keširanje podataka širom sveta, kao što je već spomenuto, idealno bi bilo korišćenje **Amazon CloudFront**-a. Kao CDN servis, **CloudFront** omogućava efikasnu distribuciju keširanih kopija podataka na edge lokacijama, značajno poboljšavajući brzinu isporuke sadržaja korisnicima širom sveta.

Strategija za postavljanje AWS Elastic Load Balancer-a

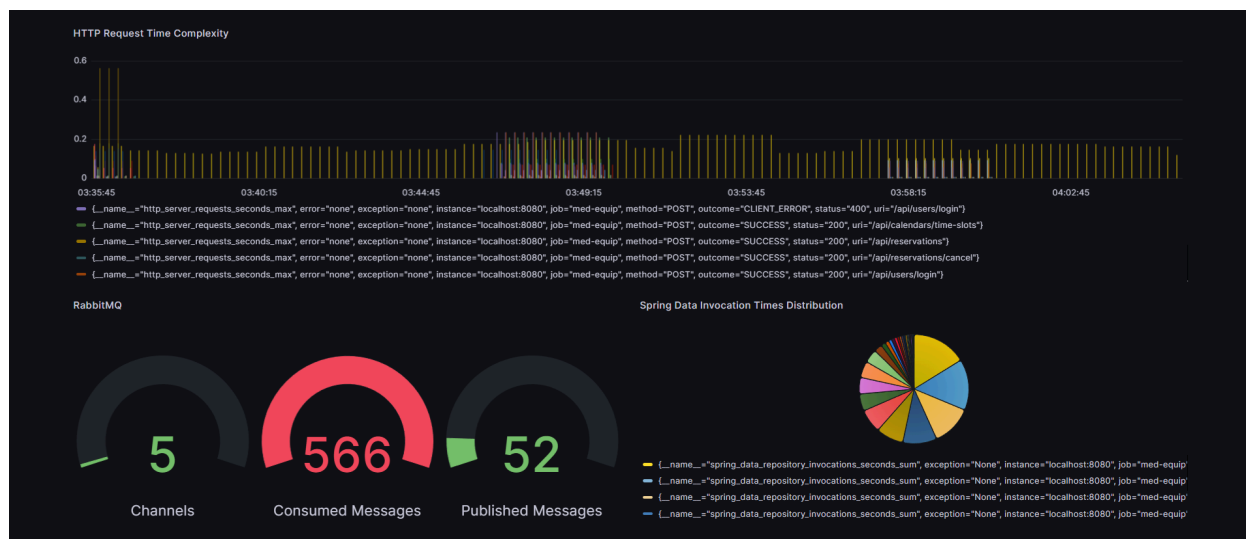
Za postizanje visoke dostupnosti, unapređenja skalabilnosti i efikasne ravnoteže opterećenja, implementacija **AWS Elastic Load Balancer (ELB)** je ključna komponenta load balancing strategije. **ELB** će automatski distribuirati dolazne zahteve između više servera, pružajući neprekidnu dostupnost usluga. Ova strategija uključuje balansiranje opterećenja na nivou transporta i aplikacije, pružajući preciznu kontrolu nad raspodelom opterećenja. Load balancer će proaktivno identifikovati i izbegavati nefunkcionalne serverske instance, povećavajući stabilnost sistema. Dodatno, optimalno upravljanje protokom podataka, različiti algoritmi ravnoteže (poput **Round Robin** ili **Least Connections**), rezervni kapaciteti i monitoring u realnom vremenu doprinose celovitosti strategije.

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

Procenu hardverski resursa nećemo raditi direktno, jer ćemo se okrenuti cloud rešenju. U početnoj fazi, planiramo distribuirano skladištenje podataka pomoću Citus čvorova na EC2 instancama, kao i još 5 EC2 instanci koje bi predstavljale servere, s namerom da dodajemo dodatne instance tokom vremena, prateći dinamiku saobraćaja po pojedinim geografskim regionima. U zavisnosti od opterećenosti regiona, dodavali bismo nove instance, ili u krajnjem slučaju uveli nove regione. Jedini otežavajući faktor može da bude ograničen izbor regiona koji AWS nudi. Ova postepena skalabilnost nam omogućava da efikasno prilagodimo hardverske resurse u skladu sa rastućim zahtevima sistema.

Nadgledanje ključnih operacija korisnika radi unapređenja performansi i korisničkog iskustva u okviru web aplikacije

Praćenje aktivnosti rezervacija termina registrovanih korisnika ima ključnu ulogu u optimizaciji performansi naše web aplikacije. Analiziranjem ovih aktivnosti stičemo dublje razumevanje ponašanja korisnika, omogućavajući nam identifikaciju trendova i potencijalnih prepreka u procesu rezervacije. Ovo praćenje ne samo da nam pruža uvid u preferencije korisnika, već nam omogućava i optimizaciju koraka u procesu rezervacije, čime se ubrzava i poboljšava korisničko iskustvo. Otkazivanje termina predstavlja još jednu ključnu aktivnost za analizu, pružajući nam informacije o nezadovoljstvu korisnika ili eventualnim preprekama sa kojima se suočavaju. Identifikacija potencijalnih problema omogućava nam brzu reakciju i kontinuirano poboljšanje sistema. Za prikupljanje metrika koristimo Prometheus, dok se vizualizaciju istih obavlja pomoću platforme Grafana.



Dizajn arhitekture

