

Javohir (Ned)

Backend dasturchi va PostgreSQL DBA | Batafsil ish tajribasi

+82 10-8864-3848 | nediro883@gmail.com | github.com/kicheol83 | KakaoTalk open.kakao.com/o/sa2bEaRi | javohir.dev | Seoul, Sodemun-gu

Tajriba

Backend dasturchi (masofaviy, Toshkent)

2024.09 - 2025.08

Sigma Software

Ukrainada asos solingan global IT xizmatlar kompaniyasi (2,000 dan ortiq mutaxassis, 12 davlat). 6 kishilik jamoada (backend 3, frontend 2, QA 1) backend API va DB sxemasi uchun mas'ul edim. Java, Kotlin, Spring Boot, PostgreSQL, Redis, Docker, GitHub Actions, AWS, Kubernetes.

- Buyurtma ro'yxati API p95 1.4 s → 210 ms: kompozit indeks qo'shildi, N+1 so'rovlar bitta JOIN'ga birlashtirildi (Grafana p95 oldin/keyin)
- Redis kesh joriy qilingach, asosiy DB'ga takroriy o'qish so'rovlari ~45% kamaydi (pg_stat_statements calls oldin/keyin)
- Streaming Replication asosidagi Primary-Standby HA sozlashda ishtirok etdim; failover 5-10 soniyada
- Yevropalik mijoz uchun buyurtma va hisob-faktura API'larini Java/Kotlin, Spring Boot'da loyihaladim, Flyway migratsiyalarini joriy qildim

Backend dasturchi va PostgreSQL DBA (masofaviy, Toshkent)

2022.10 - 2024.05

LeverX Group

SAP platformasida dasturiy ta'minot hamda web, mobil va cloud yechimlar ishlab chiqaruvchi xalqaro IT kompaniya (1,500 dan ortiq xodim). 8 kishilik jamoada integratsiya servislari, hisobot API'lari va DB uchun mas'ul edim. Node.js, NestJS, PostgreSQL, REST, Docker, Redis, AWS.

- PostgreSQL 13 → 15 major upgrade'ni logical replication orqali servisni to'xtatmasdan yakunladim
- Ichki tizimlar o'rtasida ma'lumot almashuvchi integratsiya servisini yozib, qo'lda bajariladigan ishlarni olib tashladim
- Katta hisobotlarga pagination va batch eksport qo'shib, serverni yuklamasdan bo'laklab qayta ishlashni ta'minladim
- Unit va integratsiya testlarini yozib CI'ga uladim, sekin DB so'rovlarini optimallashtirdim

Frilans

Loyiha asosida xorijiy mijozlarning DB tezligi, xavfsizligi va kengayish muammolarini hal qildim (jami 16 oy).

BAA, Ishlab chiqarish · 2021.11 - 2022.04 (6 oy)

Muammo: Ma'lumot ko'paygani sababli kun oxiridagi ishlab chiqarish hisobotlari sekin ochilib, menejerlar uzoq kutardi.

Asosiy ishlar: Sekin so'rovlarni EXPLAIN ANALYZE bilan tahlil qilib qayta yozdim, indekslar qo'shdim, ko'p ishlatiladigan murakkab hisobotlar uchun Materialized View joriy qildim.

Natija: Hisobot yuklanishi 12 s → 0.8 s (~15x); DB'dagi umumiy yuklama kamaydi

Gonkong, Restoran / F&B · 2023.01 - 2023.04 (4 oy)

Muammo: Buyurtma tizimidagi ayrim DB foydalanuvchilari keraksiz ortiqcha huquqlarga ega edi, bu xavfsizlik riskini oshirardi.

Asosiy ishlar: Ruxsatlarni RBAC asosida qayta tuzdim, jadvallarga kirishni RLS bilan chekladim, kunlik backup va recovery jarayonini joriy qildim.

Natija: Ortiqcha superuser ulanishlari 7 → 0; backup'dan tiklash testda 15 daqiqadan kam

Bangladesh, AI xizmatlari · WellDev Bangladesh Ltd. · 2025.03 - 2025.09 (6 oy)

Muammo: AI model natijalarini saqlovchi jadval tez o'sib, INSERT'lar sekinlashgan edi.

Asosiy ishlar: Jadvalni oylik range partitioning bilan ajratdim, eski ma'lumotlar uchun arxivlash strategiyasini joriy qildim.

Natija: INSERT p95 320 ms → 45 ms (~7x); aktiv ma'lumot 180 GB → 22 GB

PG Insight

Ochiq multi-tenant monitoring xizmati: har kim ro'yxatdan o'tib, o'z PostgreSQL bazasini qo'shadi va faqat o'zi qo'shgan bazalarni ko'radi

2026.08 - 2026.10

Rol: Yakka: dizayn, backend, frontend, xavfsizlik, testlar, deploy va operatsiya (81 commit)

NestJS, TypeScript, Prisma, PostgreSQL, TimescaleDB, Socket.IO, React, Vite, Tailwind CSS, Recharts, Jest, Vitest, Docker, Caddy

Nimani isbotlaydi

- Tenant izolyatsiyasi: global guard har bir :targetId route'ida egani tekshiradi, begona target'ga 403 emas, 404 qaytaradi; himoyasiz mutating route qo'shilsa test yiqiladi
- SSRF himoyasi: DNS javobidagi barcha IP tekshiriladi, ulanish tekshirilgan IP'ga qotiriladi (TLS SNI saqlanadi); production'da platform-db, 169.254.169.254, 127.0.0.1 rad etildi
- Ixtiyoriy SQL'ni cheklash: EXPLAIN 6 qatlamda himoyalangan, READ ONLY va 5 s statement_timeout bilan; pg_sleep 0.05 s da, generate_series(1, 1000000000) 5.4 s da 400 bilan to'xtatildi
- Kuzatiladigan bazaga yukni o'lchadim: 10 daqiqada 1 yadroning 0.06% i, konteyner CPU'si +0.3%
- Operatsiya: SHA-tagli deploy, health tekshiruvidan keyingina yozib qo'yish, ikki yo'nalishli rollback mashqi

Muammo

Ishlab turgan PostgreSQL'dagi sekin so'rovlar, lock kutishlari, bloat va replikasiya kechikishi turli vosita va SQL orqali alohida tekshiriladi. Ularni bitta xizmatga yig'dim; bu begona bazalarga ulanadigan ochiq xizmat bo'lgani uchun tenant izolyatsiyasi, SSRF himoyasi va ixtiyoriy SQL'ni cheklash boshidanoq kerak edi.

Arxitektura

Collector'lar har bir bazadan 3 soniya-5 daqiqa oralig'ida metrika yig'ib, TimescaleDB'ga (10 ta hypertable) yozadi; jonli sahifalar esa bazaga to'g'ridan-to'g'ri so'rov yuboradi. Foydalanuvchi, target, alert va audit Prisma 7 boshqaradigan platforma DB'da (PostgreSQL 16). NestJS 11 global guard'lari (Throttler → JWT → TargetAccess) har bir :targetId route'ida egalikni tekshiradi, target so'rovlari esa target boshiga pool'da (maks. 5) xom SQL bilan bajariladi. Rollar ikkita: admin (birinchi hisob) va user (o'z target'lari, 3 tagacha, faqat ommaviy manzillar); UI koreys, ingliz va o'zbek tillarida.

Asosiy qarorlar

- Demo emas, ochiq multi-tenant xizmat: Ro'yxatdan o'tish ochiq, har bir foydalanuvchi faqat o'zi qo'shgan target'larni ko'radi. / Evaziga: Tenant izolyatsiyasi, SSRF, kvota va rate limit majburiy bo'ldi.
- Egalik tekshiruv controller'da emas, global guard'da: Throttler → JWT → TargetAccess tartibidagi global guard'lar har bir :targetId route'ini tekshiradi. / Evaziga: Boshqa ID'li route'lar uchun @TargetAccess dekoratori kerak.
- Begona target'ga 403 emas, 404: Ruxsat bo'lmasa, mavjud bo'lmagan target bilan bir xil 404 qaytadi. / Evaziga: Admin uchun ham mavjudlik alohida tekshiriladi.
- Host'ni resolve qilib, IP'ga qotirish: DNS javobidagi barcha IP net.BlockList bilan tekshiriladi va ulanish tekshirilgan IP'ga to'g'ridan-to'g'ri qilinadi. / Evaziga: TLS uchun SNI qo'lda beriladi; o'z-o'zini hosting uchun TARGET_ALLOW_PRIVATE_HOSTS.
- EXPLAIN qoladi, lekin READ ONLY + 5 s + funksiya denylist: 6 qatlamli tekshiruvdan keyin READ ONLY tranzaksiyada statement_timeout va lock_timeout bilan bajariladi. / Evaziga: Vaqt va funksiya cheklovi sabab ba'zi to'g'ri so'rovlar ham rad etilishi mumkin.
- Yangi target uchun standart sslmode=require: Yangi target'lar va DB default'i require'ga o'zgartirildi (47287b4). / Evaziga: TLS'siz bazada foydalanuvchi disable'ni o'zi tanlaydi.
- Migratsiyalar orqaga qaytarilmaydi: Har bir migratsiya qo'shimcha shaklda (avval qo'shish, keyin tozalash). / Evaziga: Har migratsiya oldingi kod bilan mos bo'lishi kerak.

Asosiy raqamlar

- 10 daqiqada target'ga yuborilgan so'rovlar: 2 430
- Target'ga yuk (1 yadroga nisbatan): 0,06 %
- So'rovning o'rtacha ijro vaqti: 0,14 ms
- Yig'ilgan ma'lumot kechikishi: 1,4 s
- EXPLAIN ichidagi pg_sleep'ni to'xtatish vaqti: 0,05 s
- Avtomatik testlar (188 backend + 49 frontend): 237

Nosozlik

API restartdan keyin hech bir target kuzatilmadi; overhead o'lchovida PG Insight'dan 0 ta so'rov chiqdi.

Sabab: Log Starting collection for 0 active target(s) ni ko'rsatdi: orchestrator onModuleInit'da pool'lar ulanishidan oldin ishga tushgan, 14:23-17:25 UTC oralig'idagi ma'lumot yo'qoldi.

Yechim: Ishga tushish onApplicationBootstrap'ga ko'chirildi (6246ce2); restartdan keyin ikkala target ham darhol yig'iladi.

Takrorlanmasligi uchun: Xatoni qayta chiqaradigan lifecycle testi qo'shildi, u pauzadagi target'lar o'tkazib yuborilishini ham tekshiradi; yangi testlar eski kodda yiqilishi tasdiqlangan.

Test va sifat

- Testlar: 237
- CI: GitHub Actions (워크플로 준비, billing 문제로 현재 중단)
- Jest, Supertest, Vitest, 실제 PostgreSQL 16

Natija

Yig'ish kuzatilayotgan bazaga 10 daqiqada 2 430 so'rov va jami 353.4 ms ijro vaqti berdi, ya'ni 1 yadroning ~0.06% i. Haqiqiy production bazalarda (LedgerCore, FlashDrop) 5 ta, kod ko'rigi va testlarda yana 3 ta xato topildi; har biri avval test bilan qayta chiqarilib, keyin tuzatildi. Boshqa holatlar: real-time broadcast xatosi collector'larni to'xtatgani (b9825e4), lock so'rovidagi RENAME sintaksis xatosi (b9825e4), connection metrikasiga sessiya boshlangan vaqt yozilgani (d587ce6, 891 noto'g'ri qator o'chirildi), pauzadagi target restartdan keyin o'zi yoqilgani (41569c7), parol o'zgarganda SSL yo'qolgani va prefer amalda shifrlamagani (41569c7, 47287b4), birinchi rollback mashqida eski image yo'qligi sababli skript xavfsiz to'xtatgani.

pginsight.javohir.dev | github.com/kicheol83/pg-insight

LedgerCore

Pul o'z-o'zidan paydo bo'lmaydigan va yo'qolmaydigan, hech bir o'tkazma ikki marta bajarilmaydigan PostgreSQL double-entry ledger

2026.08 - 2026.10

Rol: Yakka: dizayn, backend, DB, testlar, yuk testi, deploy (53 commit)

NestJS, TypeScript, PostgreSQL, node-pg-migrate, zod, React, Vite, Vitest, k6, Docker, Caddy, GitHub Actions

Nimani isbotlaydi

- Parallellikda to'g'rilik: tartiblangan qator qulflari bilan 12/12 yuk testida deadlock 0 va fault 0; reversal race topilib, test bilan tuzatildi
- Invariantlar bazada: debet = kredit qoidasi COMMIT paytida trigger bilan tekshiriladi; yuk ostida ham, zaxiradan tiklangan bazada ham imbalance 0
- O'lchovga asoslangan optimallashtirish: balans o'qish bitta statement'ga qisqartirildi, A/B'da p95 24.9 → 13.4 ms
- Muammoni dalil bilan topish: yuk ostidagi 10 soniyalik to'xtash Postgres va API loglarini vaqt bo'yicha solishtirib topildi, sababi log yozish bo'lib chiqdi
- Operatsiya: SHA-tagli deploy, ~1 soniyalik rollback, har kungi zaxira va uni tiklab tekshirish

Muammo

Hisoblar orasidagi pul harakatini yozadigan tizimning yagona talabi — hech qachon xato qilmaslik. So'rovlar bir vaqtda kelsa ham, mijoz qayta yuborsa ham, jarayon o'rtada to'xtasa ham pul paydo bo'lmasligi, yo'qolmasligi va bitta o'tkazma ikki marta bajarilmasligi kerak.

Arxitektura

API (NestJS 10 modulli monolit) foydalanuvchi va hisoblar, depozit va yechib olish, o'tkazma, reversal, balans, tarix va reconciliation'ni bajaradi. O'sha kod bazasidagi alohida worker jarayoni outbox hodisalarini FOR UPDATE SKIP LOCKED bilan olib, HMAC imzoli webhook sifatida yuboradi. Invariantlarni PostgreSQL 16 deferred constraint trigger'lari tekshiradi, shuning uchun ilova kodidagi xato ularni chetlab o'tolmaydi. Operatsion konsol (React + Vite) hisoblar, o'tkazmalar va outbox yetkazilishini ko'rsatadi.

Asosiy qarorlar

- ADR-0001 Pul — BIGINT minor unit, API'da string: BIGINT minor unit'da saqlanadi, bigint asosidagi Money bilan o'raladi va har bir chegarada string sifatida yuboriladi. / Evaziga: Mijoz summani string sifatida o'qishi kerak; evaziga float xatosi yo'q.
- ADR-0002 Balans ustuni yo'q; balans yozuvlardan hisoblanadi: Balans har doim ledger_entries yig'indisi; narxi covering index bilan cheklanadi. / Evaziga: O'qish narxi tarix bilan o'sadi; o'lchov talab qilganda snapshot qo'shiladi.
- ADR-0003 ORM o'rniga raw SQL: pg driver va raw SQL, migratsiyalar node-pg-migrate bilan. / Evaziga: Tip xavfsizligini kompilyator emas, integratsion testlar ta'minlaydi.
- ADR-0004 Modulli monolit: accounts, ledger, idempotency, outbox modullari bitta jarayon va bitta bazada. / Evaziga: Modul boshqa modul jadvaliga to'g'ridan-to'g'ri murojaat qilmaydi.

- ADR-0005 Invariantlar deferred constraint trigger'larda: COMMIT paytidagi trigger'lar kamida 2 yozuv, debet = kredit, jami = tranzaksiya summasi va valyuta mosligini tekshiradi. / Evaziga: Xato statement'da emas, COMMIT'da chiqadi.
- ADR-0006 READ COMMITTED + id tartibida FOR UPDATE: Hisob qatorlari har doim id o'sish tartibida qulflanadi. / Evaziga: Har bir yangi yozish yo'li shu tartibga rioya qilishi shart.
- ADR-0007 Faqat outbox oylik partition, ledger partition'siz: created_at bo'yicha oylik partition + DEFAULT partition faqat outbox_events'ga qo'llandi. / Evaziga: Partition'larni oldindan yaratish operatsion vazifa bo'ladi.

Asosiy raqamlar

- Yaroqli yuk testlari (deadlock 0, fault 0): 12 / 12
- Oddiy o'tkazma throughput (50 VU): 249 req/s
- Oddiy o'tkazma p95: 228 ms
- Issiq hisob o'tkazma p95 (40 VU → 1 hisob, 409 va deadlock 0): 896 ms
- Aralash yuk throughput (30 o'quvchi + 10 yozuvchi): 884 req/s
- Balans o'qish p95 (qat'iy tezlik A/B, keyin): 13,4 ms
- Backup'dan tiklash (14/14 jadval, imbalance 0): 0,68 s
- Avtomatik testlar: 392

Nosozlik

To'liq bench'da 3 ta 500 va 3 ta 409 chiqdi, ledger esa muvozanatda qoldi.

Sabab: Postgres 04:00:52.7 da lock_timeout bergan, API esa javobni 10 soniya keyin yozgan, o'sha paytda 40 ms ichida bir guruh 500 chiqqan: Node jarayonining o'zi to'xtab qolgan. Sabab — har bir so'rovni info darajasida log qilish; Windows'da stdout pipe'ga yozish sinxron, PowerShell yo'naltirishi esa ulgurmagan.

Yechim: LOG_LEVEL=warn bilan xuddi shu stsenariyda fault 0, 409 0, throughput +37%, lock_timeout 3 s, statement_timeout 5 s va idle_in_transaction_session_timeout 10 s qulflarni bo'shatdi; 500 INTERNAL_ERROR qaytargan timeout'lar endi 503 DATABASE_TIMEOUT (retryable: true) qaytaradi (8c30dba).

Takrorlanmasligi uchun: Haqiqiy Postgres va pg-pool bilan timeout → 503 bo'yicha 3 ta test yozildi; production'da asinxron logger yoki sampling kerakligi cheklovlariga yozildi.

Test va sifat

- Testlar: 392
- CI: GitHub Actions (커밋 메시지, 포맷, lint, typecheck, Postgres 서비스로 테스트, 빌드; billing 문제로 현재 중단)
- Vitest, supertest, 실제 PostgreSQL 16 (mock 없음), k6 + bench.ps1

Natija

Yakuniy kodda 12/12 yuk testi yaroqli chiqdi: fault 0, deadlock 0, contention 0, har bir rundan keyin ledger muvozanatda. Balans o'qishni bitta statement'ga qisqartirib, qat'iy tezlikdagi A/B'da p95 24.9 ms dan 13.4 ms ga tushirildi. Raqamlarga ishonishdan oldin yana quyidagilar topilib tuzatildi: ikki parallel reversal'dan ikkinchisi 409 o'rniga 422 qaytargan race (ccee39b), .env.test gitignore'da bo'lgani uchun toza klonda testlar umuman ishlamagani (e0e6884), k6 skriptlaridagi __ITER va teardown xatolari (0f76291).

ledger.javohir.dev | github.com/kicheol83/ledger_core

Smart Farm

Issiqxona uchun real-time IoT monitoring va avtomatik sug'orish: ESP32 sensor

2026.01 - 2026.10

→ MQTT → pipeline → qoida → nasos → yana qurilmaga yopiladigan kontur

Rol: Yakka: backend, frontend, ESP32 firmware, deploy va operatsiya (171 commit, 2026-10-04 dan production'da)

Nimani isbotlaydi

- IoT ma'lumotlari yo'qolmasligini o'lchab isbotlash: MongoDB 132 s o'chganda 24/24, backend 20 s to'xtaganda 7/7; eski koddagi yo'qotish (12 dan 1, deploy'da 1) avval o'lchangan
- Xavfsizlik auditi va tuzatish: signup'da ADMIN bo'lish, ixtiyoriy maydon yozish, boshqa a'zo resursiga kirish, OTP'ni cheksiz taxmin qilish, kodga yozilgan JWT kaliti kabi 12+ zaiflik production'dagi suiiste'mol izlari tekshirilgach yopildi
- Qurilma va backend o'rtaqidagi halqani yopish: RELAY_ON/OFF MQTT orqali yuboriladi, nasos vaqti server (30 daq) va firmware (1800 s) tomonida cheklangan, avtomatik qoida bir vaqtda ikki marta ishlamaydi
- Muhandislik intizomi: 89 ta yangi testning har biri eski kodda yiqilishi tekshirilgan; noto'g'ri gipoteza va yetarli bo'lmagan testlar ham hujjatlangan

Muammo

Issiqxona sensor qiymatlari uzilib yoki sakrab turadi, qurilma vaqti noto'g'ri bo'lishi mumkin, DB yoki backend qisqa to'xtasa ham o'qishlar yo'qolmasligi kerak. Tuproq quriganda odam ekranni ko'rishidan oldin nasos yoqilishi, tarmoq uzilsa ham nasos to'xtamay suv toshib ketmasligi kerak. Boshqa a'zoning fermasiga kirib bo'lmisligi esa asosiy talab.

Arxitektura

Haqiqiy ESP32 (DHT22, capacitive tuproq namligi, BH1750 yorug'lik, suv sathi, rele + 12 V nasos) va production demo uchun simulyator bir xil yo'l bilan Mosquitto'ga (TLS 8883, QoS 1) yuboradi. NestJS 11 IoT pipeline qurilma API kaliti, payload shakli, topic ↔ deviceId mosligi, sensor egaligi va kalibrovkani tekshiradi, Redis bufer (qayta urinish, DLQ) orqali MongoDB'ga xom qiymat va 90 kunlik TTL time-series yozadi, soatlik, kunlik va oylik agregatlar quradi. Har sensor uchun Z-score anomaliya (≥ 2 warning, ≥ 3 critical), o'simlik salomatligi indeksi va Open-Meteo ob-havosi (10 daqiqalik kesh) hisoblanadi; avtomatlashtirish qoidasi aktuatorni o'zgartirsa, qurilmaga MQTT orqali RELAY_ON/OFF ketadi. Dashboard GraphQL (51 resolver) va Socket.IO orqali jonli yangilanadi, koreys, ingliz va o'zbek tillarida.

Asosiy qarorlar

- Xabar buferi: Redis list o'rniga hash + ikki zset: `iot:buffer:messages` (hash), `iot:buffer:inflight` (zset, ishlov boshlangan vaqt), `iot:buffer:retry` (zset, keyingi urinish vaqti), `iot:buffer:dead` (DLQ); `ack` — id bo'yicha `O(1)` `hdel` + `zrem`. / Evaziga: Retry mantiqi o'zimizniki — shuning uchun 10 ta test va haqiqiy redis-server bilan tekshiruv.
- Har bir o'qish uchun `done[]` ro'yxati: `Promise.allSettled` natijasidan muvaffaqiyatli sensorlar `done[]` ga yoziladi va qayta urinishda o'tkazib yuboriladi; qayta urinish asl `receivedAt`'ni saqlaydi.
- Ruxsat xatosi va DB xatosini ajratish: Faqat `UnauthorizedException` tashlab yuboriladi, qolgan har qanday xato qayta urinishga tushadi.
- MQTT doimiy sessiya + ulanish `onApplicationBootstrap`'da: `clean:false` va barqaror `clientId` (ikkinchi instansga `MQTT_CLIENT_ID` orqali alohida id), ulanish barcha `onModuleInit`'lardan keyin — `onApplicationBootstrap`'da.
- Xato logini yozish best-effort: Log yozish `try/catch` ichida, yiqilsa faqat `warn` — `retry` hisobiga ta'sir qilmaydi.
- Egalik tekshiruvi: global interceptor + `@OwnedArgs`: Global `OwnershipInterceptor` argument nomidan resursni topib, zanjir bo'ylab `farm.memberId` gacha chiqadi; ko'p ma'noli argumentlar uchun `@OwnedArgs({ 'input.fieldId': 'fieldMap' })`. / Evaziga: Har bir so'rovga qo'shimcha egalik tekshiruvi qo'shiladi.
- Nasos cheklovi ikki tomonda: `Server autoOffAt`'ni maksimal 30 daqiqaga cheklaydi; firmware `ticks_ms` asosida 1800 s + 60 s `grace`'dan keyin o'zi o'chiradi.
- Avtomatik qoida: `cooldown`'ni atomik egallash: `findOneAndUpdate` filtrida `cooldown` sharti bor — faqat bittasi qoidani egallaydi.

- recordedAt uchun ehtiyot nusxa: Qurilma vaqti 24 soatdan eski yoki 5 daqiqadan kelajakda bo'lsa, server qabul vaqtidan foydalanadi, soniyadagi vaqt ham avtomatik aniqlanadi; firmware NTP'ni har soatda qayta urinadi.

Asosiy raqamlar

- MongoDB 132 s o'chganda saqlangan xabarlar (DLQ 0): 24 / 24
- Backend 20 s to'xtaganda yetib kelgan xabarlar: 7 / 7
- IoT deploy'dan keyingi 10 daqiqada saqlangan qatorlar (rad etilgan 0): 320
- Rad etilgan xavfsizlik probe'lari: 5 / 5
- Yangi testlar (har biri eski kodda yiqilishi tekshirilgan): 89
- Nasosning maksimal yoniq vaqti (server): 30 min
- Eng uzun qayta urinish oralig'i (30 / 60 / 120 s, 3 tadan keyin DLQ): 120 s
- MongoDB kolleksiyalari: 40

Nosozlik

Signup mutation'iga memberRole: "ADMIN" yuborilsa, javob ADMIN a'zo qaytarardi (probe before).

Sabab: CreateMemberInput'da memberRole maydoni bor edi va memberModel.create(input) butun inputni yozardi; profil yangilashda MemberUpdateInput ham rol, status, email va parolni qabul qilardi. Shu auditda @Roles(MANAGER) bor, lekin @UseGuards(RolesGuard) yo'qligi sabab a'zolar ro'yxati tokensiz ochiq edi ({\"total\":7}), guard klass darajasidagi @Roles'ni o'qimagani uchun bufer resolverlari ham ochiq edi.

Yechim: Maydon inputdan olib tashlandi, create va \$set faqat ruxsat etilgan maydonlar bilan (22a760bd); guard qo'shildi, getAllAndOverride bilan handler + klass o'qiladi, noto'g'ri token → 401.

Takrorlanmasligi uchun: Tuzatishdan oldin production'da suiiste'mol izlari tekshirildi (7 a'zo, hammasi WORKER /ACTIVE, imtiyozli 0, shubhali chaqiruv 0). member-security.spec.ts'dagi 13 testdan 12 tasi eski kodda yiqiladi; deploy'dan keyin 5 ta tashqi probe'ning hammasi rad etildi.

Test va sifat

- Testlar: 89
- Jest 30, ts-jest, 실제 redis-server, Mosquitto 2.0.18, MicroPython stub harness

Natija

2026-10-10-11 auditida 9 ta tuzatish commit'i chiqarildi, har biri oldin/keyin o'lchovi bilan. MongoDB 132 s o'chirilganda 24/24 xabar saqlandi (6 qayta urinish, DLQ 0); backend 20 s to'xtatilganda 7/7 yetib keldi, backend o'chiq paytida yuborilgani broker navbatidan keldi. Eski kodda 58 s outage'da 12 dan 1 tasi, deploy restartida 1 tasi yo'qolgan edi. Xavfsizlik tekshiruvida 12+ zaiflik topilib yopildi: signup'da memberRole bilan ADMIN bo'lish (22a760bd), profil orqali ixtiyoriy maydonlarni yozish, boshqa a'zoning fermasi, sektori va alertiga kirish (c2c5ca26), OTP'ni cheksiz taxmin qilish va Math.random OTP, kodga yozilgan JWT kaliti (a5a1b62e), production'da ochiq GraphQL playground. Tuzatishdan oldin production'da suiiste'mol izlari tekshirildi (7 a'zo, imtiyozli 0, shubhali chaqiruv 0). Backend nasosni faqat DB'da yoqib, qurilmaga buyruq yubormagani (3f3eacb2) va NTP sinxronlanmagan ESP32 1970-yil vaqtini yuborgani ham tuzatildi. Birinchi gipotezam (xatolar heartbeat'dan) noto'g'ri chiqqanda log haqiqiy sababni (/sensors) ko'rsatdi; 45 s outage driver'ning 30 s kutishi sabab xato yo'lini umuman ishga tushirmagani ham hujjatlandi.

farm.javohir.dev | github.com/kicheol83/smart-farm

FlashDrop

Ortiqcha berilmaydigan navbat kuponlari: 4 concurrency usuli yozilib, k6 bilan o'lchab solishtirildi

2026.09 - 2026.10

Rol: Yakka: dizayn, backend, testlar, yuk testi, frontend, deploy va ekspluatatsiya (21 commit)

Nimani isbotlaydi

- Muammoni isbotlash: lock'siz usul 50 o'rniga 300 ta kupon berdi, remaining_quantity esa 0 emas, 46 da qoldi
- To'g'ri yechimlar: qolgan uch strategiya 300 so'rovdan aniq 50 tasini o'tkazdi, DB'da ham aynan 50 ta yozuv
- Natijani tushuntirish: bu yukda optimistic lock FOR UPDATE'dan nega tez chiqqani topildi — FOR UPDATE rad etiladigan 250 so'rovni ham navbatda kutdiradi
- O'lchov intizomi: warm-up, 3 run mediana, TCP backlog'dan qisman rad etilgan runlar avtomatik yaroqsiz; qayta o'lchov xulosani o'zgartirganda README tuzatildi
- Operatsiya: SHA-tagli deploy va rollback, har kungi xaxira va tiklash testi, conflict belgisi deploy bo'lgan holatdan keyin qo'yilgan himoya

Muammo

50 dona kuponga bir vaqtda 300 ta so'rov keladi va bitta ham ortiqcha chiqmasligi kerak. Oddiy "o'qi → tekshir → yoz" mantiq'i lost update tufayli 50 o'rniga 300 ta kupon chiqaradi. Bir xil biznes qoidasini 4 usulda yozib, to'g'rilik, throughput va latency o'rtasidagi muvozanatni real yuklama ostida o'lchash kerak edi.

Arxitektura

Caddy statik frontend'ni beradi va /api/* ni Spring Boot'ga uzatadi. So'rov avval Bucket4j rate limiter'dan o'tadi (holati Redis'da). Har strategiyaning o'z endpoint'i, entity'si va jadvali bor. Redis usulida qoldiq va "olganlar" to'plami bitta Lua skriptda atomik tekshiriladi, PostgreSQL'ga yozish @Async bilan fonda.

Asosiy qarorlar

- Redis: Lua'da DECR + SADD, DB'ga @Async: SISMEMBER → GET → DECR + SADD bitta claim.lua ichida: hot path'da Postgres ham, check-then-act oralig'i ham yo'q. / Evaziga: Redis va DB orasida ilova qulasa, kupon DB'da yo'qolishi mumkin.
- Optimistic retry ikki bean'ga bo'lingan: OptimisticClaimService (retry, maks. 20) → OptimisticClaimAttempt (@Transactional, @Version) / Evaziga: Bitta mantiq ikki klassga bo'linadi.
- Har strategiyaga alohida jadval va endpoint: To'rt strategiyaning har biri o'z campaign / claim jadvallari, entity va endpoint'iga ega. / Evaziga: Kod va migratsiyalar takrorlanadi (8 jadval).
- Lock'siz usul ataylab qoldirilgan: Lost update aynan sodir bo'ladigan endpoint saqlandi, uning jadvaliga CHECK ham qo'yilmadi. / Evaziga: Production'da bunday endpoint bo'lmaydi.
- Idempotency va qoldiq chegarasi bazada: UNIQUE(campaign_id, user_id), CHECK (remaining_quantity >= 0): Har bir claim jadvaliga UNIQUE, uchta to'g'ri strategiyaga V9 CHECK; production'da remaining_quantity = -1 yozish rad etildi. / Evaziga: Redis strategiyasida ikki joyda tekshiriladi.
- Rate limit kaliti — userId: Bucket4j + Redis (Lettuce) token bucket userId bo'yicha, bir nechta nusxada ham limit umumiy. / Evaziga: Mijoz uni o'zgartira oladi; production'da IP bilan birga kerak.

Asosiy raqamlar

- Redis Lua throughput: 873 req/s
- Redis Lua p95: 178 ms
- SELECT FOR UPDATE throughput: 188 req/s
- SELECT FOR UPDATE p95: 1 465 ms
- Optimistic lock throughput: 331 req/s
- Optimistic lock p95: 831 ms
- Lock'siz: 50 kupondan haqiqatda berilgani: 300
- Rollback buyrug'i vaqti: 1,2 s

- Backup'dan DB tiklash (11/11 jadval): 0,22 s

Nosozlik

2026-10-09 deploy'dan keyin flashdrop.javohir.dev'da UI buzildi, sahifada xom <<<<<<< HEAD matni chiqdi.

Sabab: Lokal repo GitHub'dan orqada qolgan edi. Push rad etilgach git pull merge'ida 4 ta faylda conflict chiqdi va belgilar tozalanmasdan commit qilindi. Frontend Caddy orqali git ishchi papkasidan berilgani uchun git pull bilan darhol production'ga chiqdi; backend rollback'i esa frontend'ni qamramadi.

Yechim: frontend/index.html oxirgi ishlagan commit'dan tiklandi, README'lar qayta yozildi, tasodifan commit bo'lgan IDE build natijasi (bin/) olib tashlanib .gitignore'ga qo'shildi va qayta deploy qilindi (7ac0754).

Takrorlanmasligi uchun: deploy.sh endi git grep bilan upstream'da conflict belgisi bormi tekshiradi va topsa deploy'ni to'xtatadi, merge faqat --ff-only bilan (a2dd204). Yashirin oqibat ham topildi: conflict tuzatilganda README'lar (ko/en/uz) eski sovuq run jadvaliga qaytib qolgan edi; portfolio matnini repo bilan solishtirganda topilib, qayta o'lgan natijalar tiklandi (a29c567).

Test va sifat

- Testlar: 4
- JUnit 5, Testcontainers (PostgreSQL 17, Redis 7), virtual threads, k6, bench.ps1

Natija

Lock'siz usul 50 o'rniga 300 ta kupon berdi (remaining_quantity 0 emas, 46 da qoldi), qolgan uch strategiya 300 so'rovdan aniq 50 tasini o'tkazdi va DB'da ham aynan 50 ta yozuv qoldi. Redis FOR UPDATE'ga nisbatan throughput bo'yicha 4.6, p95 bo'yicha 8.2 marta yaxshi chiqdi. Raqamlar 1 warm-up'dan keyingi 3 ta yaroqli run medianasi; muvaffaqiyatli + sold out yig'indisi 300 dan kam bo'lgan runlar (TCP backlog tufayli qisman rad etilgan) avtomatik yaroqsiz deb belgilanib, qayta yugurildi. Rate limit: bitta userId bilan 12 so'rovdan 6 tasi o'tdi, 6 tasi 429 (5 token + greedy refill).

flashdrop.javohir.dev | github.com/kicheol83/flashdrop

LiveCommerce

Jonli efir paytidagi cheklangan flash sale platformasi: 100 dona uchun 1 000

2026.09 - 2026.10

xarid urinishi, ortiqcha sotuv 0

Rol: Yakka: arxitektura, 7 ta servis, frontend, testlar, yuk testi, deploy va operatsiya (127 commit)

Kotlin, Spring Boot, Spring Cloud Gateway, Spring Data JPA, PostgreSQL, Apache Kafka, WebSocket (STOMP), Flyway, Toss Payments, OpenTelemetry, Jaeger, Next.js, TypeScript, Tailwind CSS, JUnit 5, k6, Docker, Caddy

Nimani isbotlaydi

- To'g'rilik: 100 dona uchun 1 000 urinish, 3 run — 0 ortiqcha sotuv, to'lov va sotilgan dona nomuvofiqligi 0, HTTP xato 0; har run oxirida 5 ta invariant tekshiriladi
- Nosozlikka chidamlilik: Kafka yoki to'lov bazasi haqiqatan to'xtatilganda ham 0 ta event yo'qolmaydi, 0 ta ikki marta pul yechilmaydi (chaos testlari)
- O'lchab tezlalashtirish: sotuv tugashi 1 daq 27 s → ~23 s, buyurtma yaratish p95 5.72 s → 2.68 s; connection pool gipotezasi o'lchov bilan rad etildi
- Production'dagi biznes xavfini dalil bilan yopish: bitta hisob to'lanmagan buyurtmalar bilan butun zaxirani band qila olardi; partial unique index bilan yopildi (15 → 0 o'rniga 15 → 5)
- Chegara himoyasi: login, signup va buyurtma uchun rate limit; mijoz IP'si ishonchli proxy zanjiridan olinadi, soxta X-Forwarded-For 429'dan qutqarmaydi (production'da tekshirildi)
- Operatsiya: SHA-tagli deploy, health'ga bog'langan rollback, production'da ikki yo'nalishli mashq, restartdan keyin ham token yaroqli qoladigan JWT kaliti

Muammo

Jonli efir paytidagi cheklangan sotuvda buyurtmalar bir necha soniyaga to'planadi, zaxira, to'lov va yetkazish esa turli servislarda. Flash sale'da "sotib bo'lindi" odatiy holat, shuning uchun avval pul olinsa har biri refund va PG komissiyasiga aylanadi; bitta servis to'xtasa event yo'qolishi yoki ikki marta pul yechilishi mumkin.

Arxitektura

api-gateway (Spring Cloud Gateway) JWT'ni tekshiradi, identity sarlavhalarini qayta yozadi, rate limit va WebSocket proxy'ni bajaradi. order-service'dagi Saga orkestratori avval zaxirani 10 daqiqaga band qiladi, keyin to'lov oladi (AWAITING_STOCK → AWAITING_PAYMENT → PAYMENT_CONFIRMING → CONFIRMING_STOCK → COMPLETED, to'lov oynasi 5 daqiqa). inventory (band qilish, tasdiqlash, qaytarish, muddati o'tganlarni tozalash), payment (Toss tasdiqlash va bekor qilish, Idempotency-Key), live (STOMP chat, real-time zaxira, tomoshabinlar soni), auth (RS256 JWT, JWKS, refresh token rotatsiyasi) va delivery (yetkazish, HMAC kuryer webhook'lari) har biri o'z PostgreSQL bazasiga ega va bir-birining ma'lumotini faqat API yoki Kafka event orqali o'qiydi. Event'lar har bir servisning transactional outbox'idan traceparent bilan chiqadi, shuning uchun Kafka orqali o'tsa ham butun saga bitta Jaeger trace.

Asosiy qarorlar

- Avval zaxira, keyin to'lov (hold-then-pay): Zaxira 10 daqiqaga band qilinadi, 5 daqiqalik to'lov oynasi ochiladi, to'lov tasdiqlangach hold tasdiqlanadi. / Evaziga: Zaxira 10 daqiqa band turadi, shuning uchun bir xaridorga bitta ochiq buyurtma qoidasi kerak bo'ldi.
- Choreography emas, orkestratsiya: order-service saga'ni boshqaradi, qotib qolgan qadamlarni tiklash scheduler'i har 5 s da qayta yuritadi. / Evaziga: order-service markaziy bo'g'in bo'ladi.
- Transactional outbox, broker tasdiqlagandan keyin PUBLISHED: Holat o'zgarishi va event bitta tranzaksiyada yoziladi; relay parallel yuboradi va faqat ketma-ket tasdiqlangan prefiksni PUBLISHED qiladi. / Evaziga: Dublikat bo'lishi mumkin — iste'molchilar idempotent qilingan.
- Toss chaqiruvi tranzaksiyadan tashqarida, natija outbox bilan bitta tranzaksiyada: TransactionTemplate bilan Toss tasdig'i tashqarida chaqiriladi, natija va outbox event bitta tranzaksiyada yoziladi. / Evaziga: Natija yozilmay qolsa — tiklash scheduler'i Idempotency-Key bilan qayta urinadi.
- Lock'siz skalyar o'qish bilan "sotib bo'lindi"ni oldindan rad etish: Qulfdan oldin bitta son o'qiladi va 0 bo'lsa rad etiladi. Entity o'qilmaydi — aks holda Hibernate keyingi FOR UPDATE'da eski keshlangan nusxani qaytarib, ortiqcha sotuvga yo'l ochardi. / Evaziga: Zaxira qaytarilayotgan lahzada xaridor "sotib bo'lindi" javobini olishi mumkin; ortiqcha sotuv baribir imkonsiz.
- Bitta ochiq buyurtma — partial unique index: Ochiq statuslar uchun (member_id, product_id) partial unique index (V5); buzilsa 409 OPEN_ORDER_EXISTS va mavjud orderId qaytadi. / Evaziga: Xaridor yangi buyurtmadan oldin eskisini to'lashi yoki bekor qilishi kerak; UI uni mavjud buyurtmaga qaytaradi.
- Rate limit gateway xotirasida, buyurtmalar foydalanuvchi bo'yicha: Login 10/daq, signup 5/soat IP bo'yicha; buyurtma 20/daq foydalanuvchi bo'yicha; IP XFF zanjiridagi eng o'ngdagi public manzildan olinadi. / Evaziga: Gateway bir nusxa bo'lsagina to'g'ri; login cheklovi ko'p IP'dan bitta hisobga hujumni to'xtatmaydi.
- JWT kaliti .env'da, kid — RFC 7638 thumbprint: JWT_SIGNING_KEY serverdagi .env'dan (chmod 600) yuklanadi, bir xil kalit doim bir xil kid beradi; kalit berilmasa compose start bo'lmaydi. / Evaziga: Kalit almashtirilganda eski tokenlar qisqa vaqt rad etiladi.

Asosiy raqamlar

- Ortiqcha sotuv (100 dona, 1 000 urinish, 3 run): 0
- Sotilgan = pul yechilgan: 100 / 100
- Buyurtma yaratish p95: 2,68 s
- To'lov yuborish p95: 0,63 s
- Sotuv tugash vaqti: 23 s
- Zaxira yangilanishi yetib borishi (500 tomoshabin): 99,2 %
- Tomoshabinlar soni xabarlari kamayishi: 98 %

- Chaos testlari: yo'qolgan event / ikki marta to'lov: 0

Nosozlik

2026-10-09 production'da lc-hoard-check.sh bitta test akkaunt bilan ketma-ket to'lanmagan buyurtmalar yaratdi: 2×201 , zaxira $15 \rightarrow 0$. Ya'ni bitta foydalanuvchi 10 daqiqalik hold davomida jonli efirdagi barcha mahsulotni boshqalardan yopib qo'ya olardi.

Sabab: Saga har bir POST /api/orders uchun alohida hold ochardi, "bitta xaridor — bitta ochiq buyurtma" qoidasi hech qayerda yo'q edi.

Yechim: V5 migratsiya — ochiq statuslar (AWAITING_STOCK, AWAITING_PAYMENT, PAYMENT_CONFIRMING, CONFIRMING_STOCK) uchun (member_id, product_id) partial unique index. Controller oldindan tekshiradi, poygada esa DataIntegrityViolationException index nomi bo'yicha ajratilib, 409 OPEN_ORDER_EXISTS va mavjud orderId qaytadi; frontend shu buyurtmani davom ettiradi.

Takrorlanmasligi uchun: Lokal Postgres'da ikki parallel tranzaksiyadan ikkinchisi 1.5 s kutib rad etildi; production'da qayta tekshiruv: $1 \times 201 + 409$, zaxira 5 da qoldi. 10 ta parallel so'rovdan aynan bittasi 201 ekanini tekshiruvchi integration test qo'shildi; k6 flash-sale.js endi har bir VU'ga o'z foydalanuvchisini beradi ($USERS \geq VUS$ tekshiriladi).

Test va sifat

- Testlar: 46
- JUnit 5 (توھم 24, كاوس 3, gateway دانو 14, SigningKeys 5), 실행 중인 전체 스택에 실제 HTTP, k6 3종

Natija

100 dona uchun 1 000 ta xarid urinishi (200 VU) 3 marta yugurildi: har safar ortiqcha sotuv 0, to'lov va sotilgan dona o'rtasida nomuvofiqlik 0, HTTP xato 0; har run oxirida 5 ta invariant admin API orqali tekshirildi. O'lchov bilan to'siqlar topilib, sotuv tugash vaqti 1 daq 27 s dan ~23 s ga tushdi, connection pool gipotezasi esa o'lchov bilan rad etildi. Production'da bitta hisob 2 ta to'lanmagan buyurtma bilan butun zaxirani ($15 \rightarrow 0$) band qila olishi partial unique index bilan yopildi ($15 \rightarrow 5$); auth restartidan keyin bir xil token turli servislarda turlicha javob olgan JWT kaliti muammosi (401/200) tuzatildi (0eeaf2c); login va buyurtmaga cheklov qo'shildi (085ded0). Shuningdek broker yo'qligida outbox tranzaksiyasi daqiqalab qulf ushlashi (881afda), partitsiya ko'paytirilgandan keyin consumer'lar yangi partitsiyalarni ko'rmagani, oldingi run'dagi hold muddati tugab sotilgan 101 ko'ringan k6 invariant xatosi va boshqa loyiha konteyneri 8080 portni egallagani sabab 14 ta integration test yiqilgani ham sababi topilib hal qilindi.

live.javohir.dev | github.com/kicheol83/livecommerce-saga