

Load Test Results — 2026-02-18

Load Test Results — Drop Staging

Dato: 2026-02-18 **Verktøy:** k6 v1.6.1 **Mål:** https://drop-staging.fly.dev **Server:** Fly.io shared-cpu-1x (256MB RAM, 1 delt CPU, Stockholm)

Testoppsett

To scenarier kjørt samtidig:

Scenario 1: Public Stress (helse + valutakurser)

- Ramper fra 0 → 200 samtidige brukere over 2m40s
- Ingen autentisering, tester rå serverkapasitet

Scenario 2: Autentisert brukerflyt

- Ramper fra 0 → 30 samtidige brukere
- Login → Dashboard → Transaksjoner → Mottakere → Profil
- JWT token gjenbrukt per VU

Resultater

Samtidige brukere	Median latens	p95 latens	Feilrate	Status
1-10	74ms	~90ms	0%	Fungerer utmerket

Samtidige brukere	Median latens	p95 latens	Feilrate	Status
25-50	~500ms	~3s	~5%	Degradering starter
75-100	~2-3s	~6s	~30%	Alvorlige problemer
150-200	3s+	27s+	47%	Praktisk talt nede

Detaljerte tall (k6 output)

Public endpoints:

- `health_duration`: avg=1134ms, min=54ms, med=74ms, max=44s, p90=3.5s, p95=6.2s
- `rates_duration`: avg=1077ms, min=55ms, med=74ms, max=45s, p90=3.3s, p95=6.4s
- `/api/rates` feilet i 95% av forespørslene ved høy last
- `/api/health` holdt (alltid 200)

Autentiserte endpoints:

- `http_req_duration (auth_flow)`: med=3.26s, p90=14.8s, p95=27.6s
- Dashboard og transaksjonshistorikk hardest rammet

Totalt:

- 12,841 HTTP-forespørsler over 2m53s
- 74 req/s gjennomsnitt
- 48.4% av alle forespørsler feilet

Breaking Point

~25-30 samtidige brukere

Etter dette eksploderer responstidene og endepunkter begynner å feile.

Flaskehalsen identifisert

1. Maskinressurser (KRITISK)

- `shared-cpu-1x` = 256MB RAM, 1 delt CPU
- Bokstavelig talt den minste Fly.io-planen
- CPU-metning ved ~50 samtidige forespørsler

2. SQLite single-writer (HØY)

- SQLite WAL-modus hjelper med samtidige lesinger
- Men ALLE skrivinger (rate_limits, sessions) er serialiserte
- Under last: skrivelås blokkerer lesinger

3. Null caching (MEDIUM)

- Ingen Redis eller in-memory cache
- Valutakurser hentes fra DB på hver forespørsel
- Brukersesjoner valideres mot DB hver gang

4. bcrypt 12 rounds (MEDIUM)

- Passord-hashing koster ~300ms CPU per innlogging
- Saturerer delt CPU raskt under innloggingsbølger

5. Enkeltinstans (MEDIUM)

- Ingen horisontal skalering
- auto_stop_machines = stop → kaldstarter (3.8s første forespørsel)
- min_machines_running = 0 → ingen alltid-på instanser

Oppgraderingsplan

Oppgradering	Effekt	Kostnad
Fly.io performance-1x (2GB RAM, 1 dedikert CPU)	~3x kapasitet (~75 brukere)	~\$30/mnd
+ PostgreSQL i stedet for SQLite	Samtidige skrivinger, connection pooling	~\$15/mnd
+ Redis cache (kurser, sesjoner)	10x raskere på lese-endepunkter	~\$10/mnd
+ 2 instanser (auto-scale)	~150+ brukere	~\$60/mnd totalt
Full produksjonsoppsett	~500+ brukere	~\$100/mnd

Konklusjon

For MVP/demo med SpareBank 1: Nåværende oppsett holder 10-15 samtidige brukere — tilstrekkelig for demo. For pilot med ekte brukere trengs minimum PostgreSQL + større maskin.

Se også: [Cloud Migration Strategy — GCP → Azure](#) for migreringsplan.

Revision #2

Created 2026-02-18 19:11:15 UTC by John

Updated 2026-05-24 20:01:14 UTC by John