Netanel Haber

netanel.t.haber@gmail.com

0542391003

linkedin - github - stackoverflow

2022 – 2023 Deci.Ai Fullstack/MLOPS – Python Kubernetes, React

Computer-vision platform for model benchmarking/optimization. We were expected to take full ownership of features - from the infra and devops side (Terraform, Kubernetes, ECR, ArgoCD, Argo Workflows, a whole bunch of YAML), through our Python FastAPI backend and Mongo database, to our React app.

- **Utilizing TensorRT Timing Cache in large scale (500-5000 models) experiments.** Consulting with the Algo team, I solidified the necessity of not sharing caches between nodes. Addressing complex constraints: Balancing between retaining nodes for cache persistence and the cost of maintaining idle nodes. Implemented a flexible API enabling users to control cache use and idle machine count for specific experiments. Sustaining Nodes, via 'dummy' pods using Kubernetes jobs - and pinning experiment pods to these nodes while still allowing non experiment pods on the nodes to preserve non-experiment priority. Conducted a small-scale caching experiment.
- **Hosting on-demand Tensorboard servers** for user's training experiments in our Kubernetes clusters – implemented using Kubernetes Jobs. It creates a pod that serves the Tensorboard, fetching the files from S3. The job is injected with dynamic properties using Jinja, applied within our backend using the Kubernetes Python Client, served by proxy through FastAPI into an iframe on the React side (injecting CSS&JS into the Tensorboard for customization). So, at the click of a button, the user gets access to a visualization of his training process.
- **Migration of our in-house Auth to use a SAAS service** in a time-sensitive manner. This allowed us to fix long-standing pentest issues robustly, and will allow implementing more sophisticated AuthN/AuthZ features in the future.
- **Performance - Investigating backend crashes:** Involved looking at Sentry and ElasticAPM profiling, understanding that the problem is time spent de/serlializing large amounts of unnecessary data (90%). Implemented a complex Mongo aggregation to filter dead data out. No crashes involving this request have happened since.
- **Migrated our React data management from Redux to React-query**, to address live updates, state consistency and to minimize fetches.

2020 – 2022 CardLatch Frontend Lead – React

The lead frontend dev for 2 years, the only one for 1.5 years.

- **Realtime CRUD UI**: Turned a slow read-only static dashboard to a live async read-write application, with minimal- blocking and data refresh, for a smooth user experience. **WebSockets**, **react-query**.
- **2d facility navigator (maps app)**: Basically a zoomable, navigatable canvas built from scratch - Users can upload maps, and place alarms, doors, cameras, and so on. This feature especially was used to sell the product to new customers.
- **Camera streaming app** – multiple (**16 in one tab**) security-cameras streaming, including playback. WebSockets.
- **Express proxy** that served the static artifacts, and connected to **Redis** for security-camera streaming.
- **Contributed to Spring (Java) backend.**

Management (6 months, 1 dev): Interviewed, hired, onboarded and **managed a developer**. Reviewed and merged all React and React Native code. Managed the creation of a React Native complementary app from scratch.

Cleanup and safety: Migrated codebase to TypeScript (95% of the code), Added Jest unit tests to a codebase with no coverage.

Performance: Troubleshooting in-house and library perf bugs and improving rendering and JS perf **by 1.5-10X** using primitives such as React.memo and debouncing, by drilling into the **Chrome Profiler**. Improving page and data loadtime by statically gzipping the JS and CSS bundles, dynamically gzipping JSON payloads. Continuously analyzing bundle-size for smaller lib alternatives.

Freelance and Fun

- checkers – vanilla TypeScript PVP/PVC checkers that I continuously improve for fun, with web workers, brute-force ai, minimal dom manipulation, drag&drop, mobile support, game state link serialization, undo/redo stack, etc. Vite for bundling and minifying, otherwise no libraries.
- 8086/8 subset dissassembler, written in Deno, because why not.
- netanel.dev - Personal blog and profile, written in Nuxt (Vue), with a little Tailwind sprinkled on top. Deployed to github pages.
- use-easy-infinite-scroll – npm, react infinite scrolling, used in our production app.
- vanilla-jsx – use JSX inside vanilla JavaScript, using webpack.
- Contributing to open-source libraries, like zxcvbn, react-use-websocket and usehooks, and datree.
- Android app - Pinpoint your location using the closest bus station. The app can open applications such as Moovit and Waze with the established location using the deeplinking interface.
- "Aur's dojo" - Mentoring a person with the goal of landing him his first frontend job. Includes building a plan, monitoring progress and answering questions in weekly/bi-weekly meetings.
- Metallurgy lab automation: Replaced a terrible error-prone Excel-based pipeline with C# and .wpf.

Military Service

2016 – 2019 - Unit 81 (Technological Unit, Intelligence Corps): A Metallurgy lab technician, I trained under, and eventually alongside, experienced and seasoned materials engineers. As I progressed, I received more responsibilities meant for engineers such as conducting failure analyses and writing an extensive report on the industry state of metal 3D printing.

Education

- 2019 – 2020: 1 year of hands-on training at Hyperactive, majoring in fullstack development, primarily React and Node.
- Psychometry: 720. High school Average: 120.

Languages:

Hebrew, English: Native (American-Israeli citizen).