Nitzan Raz

Software and Infrastructure Engineer

cv@backslasher.net
Backslasher.net

EXPERIENCE

AngelSense, Tel Aviv - *Chief Software Architect*

2023 -

- Working with Leadership
  - Letting VP R&D focus on owning the dev/product flow, by taking over ops.
  - Engaging with dev team, assisting with planning, code review, troubleshooting.
  - Involved in prioritization and implementation of strategic software features.
- Special projects
  - Exploratory coding and Proof-of-concept of new features to map out potential roadblocks.
  - End-to-end ownership - defining the need, planning and time. estimation, execution and implementation of monitoring.
- Tech debt elimination
  - Upgrading software infrastructure in-place (server OS, Cassandra to DynamoDB, Java version) during the normal release cycle.
  - Implementing configuration as code (e.g. Terraform, Jenkins JCASC).
  - Database version upgrades, table grooming, restructuring without affecting uptime.
- Developer Quality of Life
  - Reduced build time from 30 to 3 minutes (via big ephemeral build machines).
  - Implemented Per-commit unit test execution, handling flaky tests, improving build success from ~70% to ~97%.
- Firefighting
  - Driving the process - identifying remediating and post-mortem.
  - Communicating with non-tech stakeholders.

Facebook, Tel Aviv - *Software Engineer*

2016 - 2022

- Diverse problem finding and solving (see below), focusing on reliability, performance, scale and security (as opposed to new features).
- Worked with both user-facing teams and internal teams.
- Took part in the team's long-range planning:
  - Defined goals ("how do we measure reliability when we don't always get crash reports?") and target metrics.

    - Translated goals into projects (“How can we improve developer experience?”)
    - Split complex projects into bite-sized tasks (e.g. breaking “releasing faster” into “what unit tests are we missing to eliminate manual QA?”)
    - Onboarded and mentored team members, and helped match colleagues to projects they will enjoy and excel at.
- Consistently delivered results when collaborating with other teams (e.g. the team maintaining FB’s load balancing software, where FBLite’s TCP sessions required a lot of customization).
- Acted as a technical go-between and enabler for efforts that were not directly related to my team (e.g. PoC sending payments in Whatsap) and part-time project manager.
- Started the Infra task force presence in 2 software engineering teams
    - Talked with team leads on how PE can help (“We have no idea how many servers we’ll need next year”) and built projects out of it (e.g. create capacity measurement, acquire user growth predictions, calculate buffering, optimize regional deployment)
    - Built the oncall rotation, both technically (tooling) and procedure-wise (“what should we wake people up on? What 3am guidebooks do we need?”)
    - Created a clear crisis management methodology (what is considered a crisis? How quickly should we respond? What teams should we escalate to?)
    - Integrated PE into the team’s work routine (“let us help with AWS problems”, “we’d like to be in design review to ensure robustness”) to maximize PE’s positive influence on the team’s wellbeing and progress

Proudest moments:

- Unblocked a company-wide public launch, 2 days after joining the team.
  User Care discovered an encryption gap in their kubernetes logging flow that has to be solved by end of day. Solution involved Linux STDOUT hijacking and an uncommon encryption scheme.
- Rewrote a week-over-week engagement monitoring system to unblock a big migration of business logic outside of the main server binary, making those servers unable to report on user engagement.
  I moved the reporting mechanism upstream by using a new data platform that encompassed all of the engagement data, and adapted the platform to support real time monitoring (rather than daily-resolution aggregations).
- Migrated the billion-user-serving infrastructure of Facebook Lite from a few big servers to many tiny servers.
  Process was nontrivial as the executable was Java with a very large in-heap cache, and running it unmodified reduced user capacity dramatically.
- Crucial part of the password-logging crisis initial response.
  http://bitly.ws/wIKp for a report on the incident

- Saved 760 monthly hours of server time in 6 LoC diff.
  Colleague wanted to rewrite the entire e2e testing system as it was taking "too long" to test every commit, Rewrite was expected to take 2 months.
  I instead moved the client and server builds to run in parallel, and the rewrite plan became unnecessary, allowing the team to invest their time in more impactful projects.
- Wrote and fully delivered the "mark notification as read" feature on Facebook's website.

Dynamic Yield, Tel Aviv - *DevOps*

2014 - 2016

- Single and first DevOp in startup company
- Moved company servers from pets to cattle
  - Configuration management (Chef)
  - Servers used to be clones of each other, now set up from custom templates
  - Central log collection
  - No more SSHing into servers by devs for routine operations / deployment
  - Auto scaling groups
- Heavy custom CI/CD automation
- Chef contributor
- Security, monitoring, tuning
- Big data ops

Mamtam (Israeli Navy) - *DevOps / Team Lead*

2008 - 2014

- Sharepoint / DotNet / MsSQL DevOps
- Active Directory / Exchange / SCCM / HyperV admin
- Led a team of 9
- Commended for excellence

EDUCATION

Open University, Israel - *Bachelor of Computer Science (**GPA 93**)*

2010 - 2016