

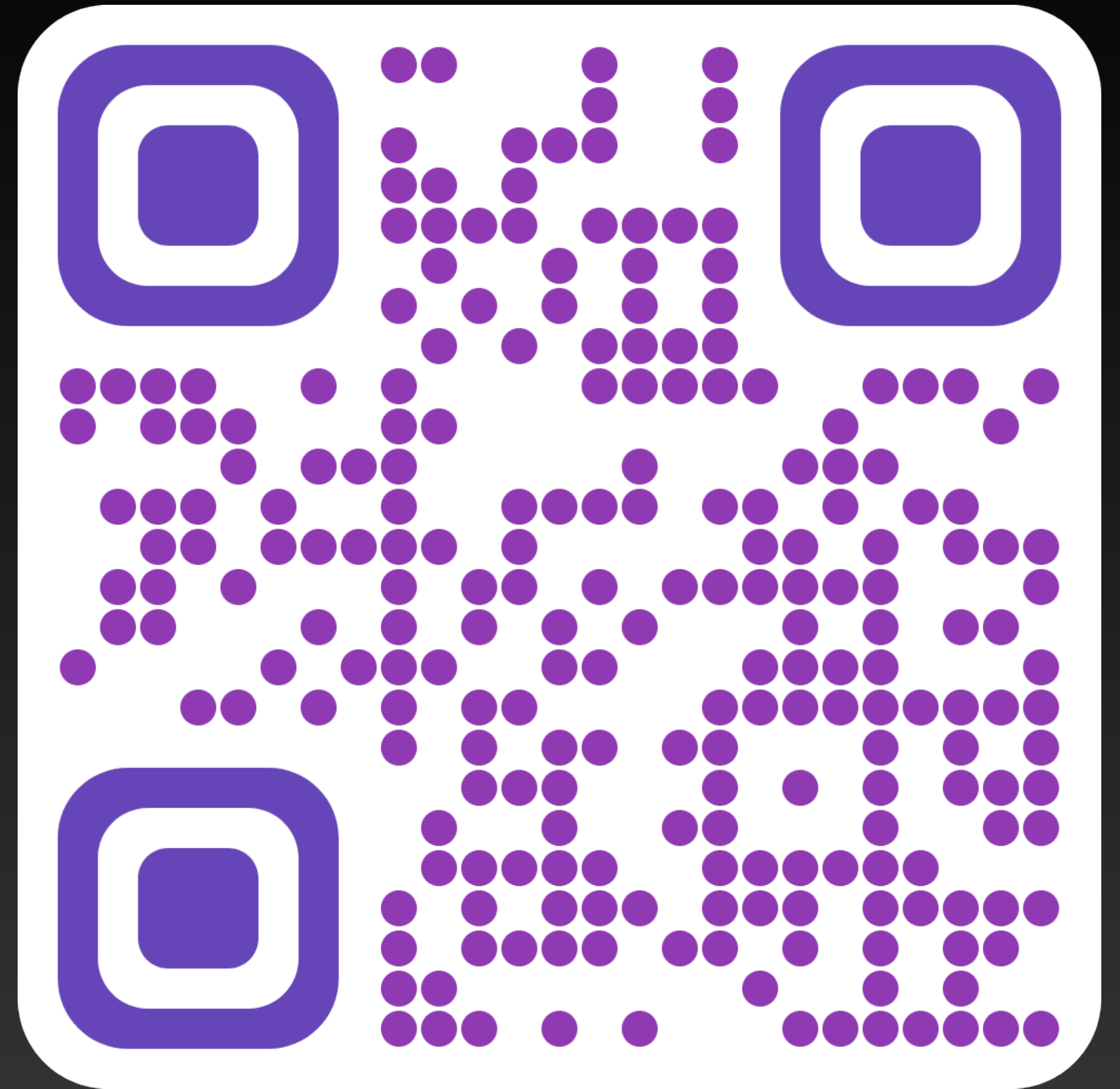
Building platforms that “Just Work”™

Learnings from getting infra from 0 to 1

Amitosh Swain Mahapatra

\$ whoami

- 🧑💻 Computer Whisperer - *I make things with computers*
- 🧑🔧 <T> Engineer at *Topolyne*
- I draw & paint 🎨
- Like to travel around with a bicycle 🚴



🔍 Scan to spot me on the internet 🌐

Building a rock-solid platform is important

Make a strong foundation

The foundation makes or breaks a sky-scraper

- Ensure extensibility and long-term scalability
- Resist the urge to adopt buzzwords, keep it stupid simple
- Prioritise best-practices, reliability, security, and performance from the beginning
- Use the tool that does the job

Shift left

- Provide platform, not implementation.
- Empower devs to take control of the CI & release system
- Introduce testing and security early in the development process



Abstract the right pieces

- Devs care about running their code reliably
- Provide just enough abstraction to customise for every service's needs without needing to tune a lot of knobs
- Identify core components that need abstraction for flexibility.
- Don't overcomplicate – only abstract what's necessary for your goals.

Centralised, yet customisable

- Create a centralised system for consistency. (Centralized helm-charts, terraform modules)
- Allow customisation where it's essential for individual team needs.
- Build simple wrappers over tools to fit your use-case

Enable first, own second

No one cares about XYZ until it runs

- The primary value of platform teams is to run stuff reliably, stick to it
- Leverage third-party tools and services when they meet your requirements
- Only build in-house solutions if there's a clear advantage or necessity
- *If no one other than you talks about the platform, you have achieved perfection*

Everyone wants to build a PaaS

But don't build your Heroku

- It's a common desire among startups to create a Platform as a Service (PaaS) to handle their custom requirements for running services
- DevOps wants to standardize everything, while product doesn't want to worry too much about right infra
- Building a reliable platform is actually building your just enough PaaS to run everything smoothly

Encourage contribution

Sharing is caring

- Platform is shared, its not something a team “owns”.
- Do not gatekeep outside contributions, in fact, encourage it
- Inner-source platform code
- Encourage team members to contribute to the platform's development and improvement

Questions?

Feel free to reach out

twitter.com/recrsn

linkedin.com/in/amitosh-swain



 Scan to spot me on the internet 