CANARY IN THE INTERNET MINE

BROOK SHELLEY
WHO IS THIS TALK FOR?

FOLKS WHO:

- make applications or websites that you visit in a browser.
- launch a new versions - even a minor one.
- need to respond to a downtime page.
- ever make mistakes.

THIS IS FOR YOU
EVERYONE SHOULD BE ABLE TO RELEASE CODE.

If a dev deploys code on your infrastructure, but no one can see it, does it matter?
IT'S COOL, THERE'S A CHANGELOG
*Please note no birds were harmed while writing this talk

CANARY? YES, PLEASE.
**TERMS**

- **Blue/Green:**
  - two production environments as identical as possible. Usually, only one is live at a time.

- **Canary:**
  - Testing term originally referring to live canaries used to check for dangerous gasses in mines.
  - In tech: server/service/deploy dies, there’s problems.
CANARY

- A small portion of live traffic.
- Testing in production.
- If the release dies, abandon that route.
CURRENT STATE

- Your app is a monolith or maybe microservices
- QA & Testing are expensive.
- Things still break.
- Rolling back your code takes 🕒 & costs 🤑.
99 SERVICES, 99 PROBLEMS?

- Each service or endpoint runs as a microservice.
- Incremental release of your application is possible.
- Combine with canary releases for success.
- Release or roll back service by service.
Early.

Often.

Customer focused metrics.
INDIVIDUAL SERVICES

- Per-endpoint/per-service testing.
- Measure often - metrics.
- Take immediate action.
- Stop ruining weekends.
HOW DO WE GET THERE?
IT SOUNDS HARD.
PROXIES & ROUTING.
Blue/Green Releases

CUSTOMERS

http://api.me.com

TURBINE LABS PROXY
DEMO TIME.
WRAP-UP

- Per-endpoint/per-service releasing.
- Measure often with user-centric metrics.
- Routing makes roll-back easy and quick.
- Stop ruining weekends.
- Developers triage at their leisure.
THANKS!

@brookshelley
brook@turbinelabs.io