Chaos, AB Testing, Toggles

Arnav Gupta

April 22, 2024

Contents

1 Chaos

Chaos is expensive in the real world but can also be free and automated in the cloud.

Chaos engineering process:

1. define steady state (baseline for normal behaviour)
2. hypothesize that steady state will continue in control and experimental group
3. introduce variables (real world events)
4. try to disprove hypothesis by looking for difference between control and experimental group

Chaos monkey randomly disables production instances to make sure system is robust enough to survive failure without customer impact.

Run chaos monkey with engineers ready and build automatic recovery mechanisms.

Other chaos monkeys are latency (artificial delays in communication), doctor (health checks), janitor (unused resources), conformity (instances that don't adhere to best-practices), security (violations/vulnerabilities), gorilla (outage of entire availability zone).

2 AB Testing and Hypothesis Testing

Toggles and field trials have different versions, where some users see each.

With logging, measure, and use for analytics to compare versions to determine if significant difference.

Statistical hypothesis testing has a null hypothesis (no difference in A and B), run stastical test, and if p-value at most 0.05, reject null hypothesis. Can get magnitude of difference.

Use RNG to determine which users see each features. Ned a statistically significant sample in each condition, and if multiple hypotheses, do a correction. Gradually increase number of users seeing a new feature.

Once enough of the new option have been seen, run a test. Use caution, with a large number of results, small differences can be statistically significant, and need to depend on effect sizes to know the difference.

3 Feature Toggles

Trunk-based development can have issue if feature is integrated and not ready for release.

Branches lead to merge hell and are hard to revert.

For feature toggles, must have conditional block, switching mechanism, and configuration file.

Critical technology for DevOps is AB testing and field trials (X% of users), darklaunching and canary release (0%, 50%, 100%), and feature rollbacks.

If error after release, toggle off the new feature, no need for new build.

DevOps is tough without trunk-based development.

Old features and new features are in the same build, but toggles control which one is displayed. Features are more isolated since they must be behind toggles.

Advantages of toggles:

- reconcile rapid release and long term feature development
- flexible feature rollout
- enable fast context switches
- features designed to be toggleable

Issues with toggles:

- toggle debt (old code is still compiled and live)
  - can't remove toggles until sure new feature/code is working
- combinatorial feature testing (features can change while system is running)
  - to deal with this, test and gradually roll out any feature combo that will be used in production and don't use combinations in production that haven't been tested or rolled out

Types of toggles are dev utils (debugging, testing), release (feature rollout), and long term (business rules).