Chaos engineering/testing

Chaos engineering

the discipline of experimenting on a distributed system in order to build confidence in the system’s capability to withstand turbulent conditions in production*

→ it’s empirical, not formal

→No models to understand what the system *should do*

→Experiments to learn what it *does*.

*** Michael T Nygard, Release It!, 2nd Edition, 2018**

"Never let a good crisis go to waste"

— Winston S. Churchill

Whole system approach

Many problems only reveal themselves in the whole system (for example, excessive retries leading to timeouts, cascading failures, dogpiles, slow responses, and single points of failure, to name a few).

→Does this mean we need to do chaos testing?

→Any other forms of testing also useful?

Netflix

Probably the best known example of chaos engineering is Netflix's "Chaos Monkey."

→Every once in a while, the monkey wakes up, picks an autoscaling cluster, and kills one of its instances. The cluster should recover automatically. If it doesn't, then there's a problem and the team that owns the service has to fix it.

→Netflix needed failures to happen *more often* so that they became totally routine.
(This is an example of the agile adage, "If something hurts, do it more often.")

https://github.com/Netflix/chaosmonkey

Injecting chaos

- Killing instances of micro services
- Kill a virtual machine
- Add latency to calls
- Drop the network
- Drop the GPS signal
- …

This is not only software related!

It’s important to study the situations where faults happen without failures. The system did something to keep that fault from becoming a failure. We should learn from those happy outcomes, as well as the negative ones.

Vulnerability

- Once you find a vulnerability
  - Find other potential instances of it
  - Think of automating the chaos that lead to the vulnerability for later testing

Chaos among people

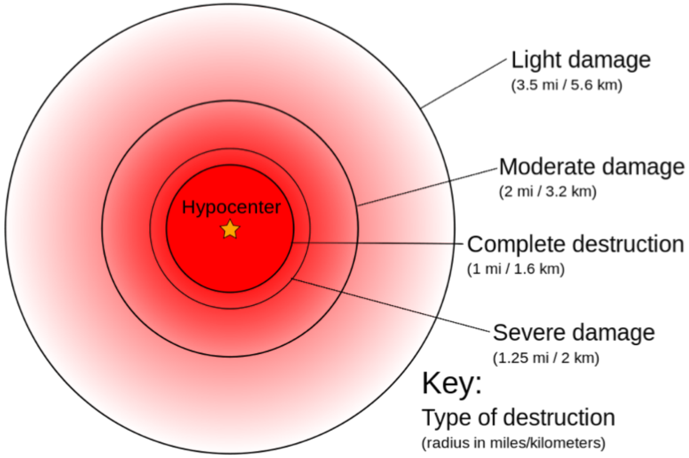
- Zombie approach
  - Randomly select 50 percent of your people and tell them they are counted as zombies for the day. They are not required to eat any brains, but they are required to stay away from work and not respond to communication attempts.
- What happens? Think truck factor!

Post-mortems

- Always do a post-mortem, trying to analyse what happened, how it was solved, whether it can happen again, whether a test or procedure should be put in place, etc.

Minimize Blast Radius

Experimenting in production has the potential to cause unnecessary customer pain. While there must be an allowance for some short-term negative impact, it is the responsibility and obligation of the Chaos Engineer to ensure the fallout from experiments are minimized and contained.



amazon
webservices™
CHAOS
MONKEY