Why does Capital One test in production?

Brought to you by:

Bryan Pinos
Sr. Director, Software Engineering
Capital One

Yar Savchenko
Director, Software Engineering
Capital One

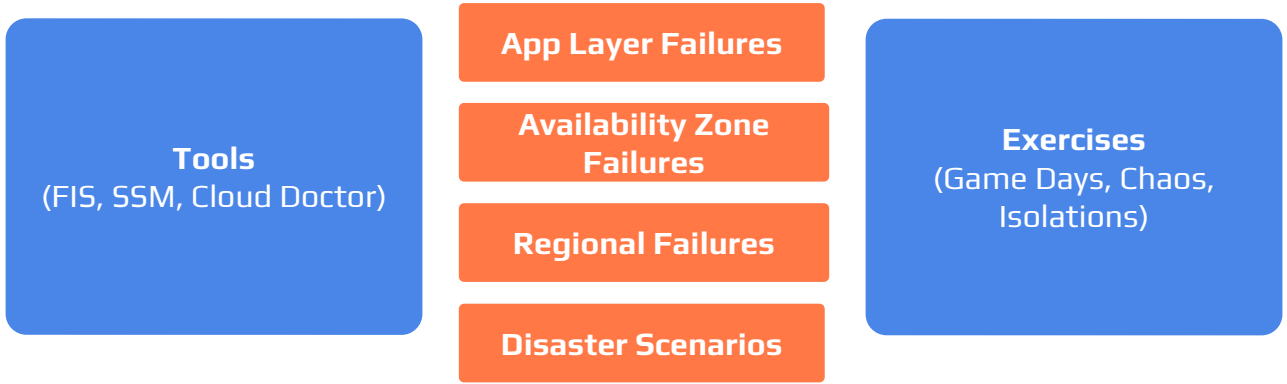
A little about Capital One

- The first bank running entirely on the public cloud
- 25-year-old, founder-led public company
- Top 10 bank and credit card issuer, and second largest FI auto loan originator
- 100M+ customers and 50,000 associates
- All-in on the cloud and one of the largest AWS users

Capital One tests in production, but why???

- Testing in production is considered a “bad word” in the tech industry and rightfully so…
- Yet Capital One does it on a regular basis and not afraid to admit it!
- **Why are we doing it?**
  - Because QA and production environments never, ever match and it is very difficult to generate full user load across all of the mico-services in QA.
- **How are we doing it?**
  - By utilizing internally developed and industry chaos engineering tools and Game Day chaos exercises.

Tool-up to fight complexity, and assume failure



Standardize deployment and **embrace IaC**

Invest in tooling to **understand** complex cloud state and call flows

Root out manual intervention through targeted **exercises**

Benefits realized via chaos engineering

Latency findings

- Proactively identified a number of potential increased latency scenarios with multiple microservices.

Capacity findings

- Proactively identified a number of microservices that were not sized correctly to handle increased volume.
- Real time measurement of key systems performance under extreme load

Outcomes

- Many of these findings have been successfully mitigated in 30 days or less, through configuration changes, capacity expansion, and re-architecture.

Are there risks? Yes, but....

Inherent risk

- Testing in production with real customer traffic always carries some amount of risk.

Unexpected impacts

- Latency, lack of adequate capacity, or actual failures; all are unpredicted and can occur at any time. Especially during planned testing.

Mitigation techniques

- Defined and agreed upon roll-back triggers and techniques that can be executed in under 5 minutes.
- Real time monitoring of all critical systems and transactions before, during, and after the Game Day exercise.
- Dedicated and experienced SRE engineers on standby and ready.

What’s next?

Scope expansion

- All critical applications across domains
- 3rd party vendors
- Execution on a highest traffic volume days with no advanced notice

Chaos engineering integration

- Integration of chaos experiments (planned failures) in production during the exercises
- Generation and injection of “fake” http error codes
- Validation and testing of automated recovery techniques and solutions

Unannounced and executed by a single team

- End goal is to execute these Game Day exercises unannounced, with a single click while utilizing automated recovery solutions…
- Then…. Achieve higher resilience!

Thank you!