Introducing Chaos Engineering as part of DevOps
What is Chaos Engineering?
What?
Who am I?

Twitter: @GurtejPalSingh  
LinkedIn: https://www.linkedin.com/in/gurtejpalsingh/  
Blog: https://gurtejpsingh.wordpress.com/  
Associate Director – Agile Service Delivery @ LoyaltyOne

Enterprise Digital Strategy (Agile Coaching & DevOps Consulting) | Planning & Execution | Technology Consulting
Agenda

• What?
• Chaos Engineering VS Testing
• The Story of How & Why
• Chaos Engineering & DevOps
• Industry Presence
Ideal Architecture

- Auto Scalable
- Fault Tolerant
- Highly Available
Let us see...
Then to now...

Monolith vs Microservices

Pic courtesy: https://blog.knoldus.com
Let us see.....

Micro Service 1

Micro Service 2

Micro Service 3

Micro Service 4

Micro Service 5
Chaos Engineering VS Testing
Chaos Engineering VS Testing

\[ (x^2 y + \sin^3 y + \cos^3 x + 2y^2 \cos^2 y) = 0 \]

\[ \text{ans:} \]

\[ \frac{10}{10} \times 0 = 0 \]

\[ \text{L.H.S.} = \text{R.H.S.} \]

\[ \text{hence, proved} \]
Models of the two...

Testing == Validation

Chaos Engineering == Experimentation
The story of How & Why...
Prerequisites

- Review the application architecture to identify:
  - Dependencies
  - Customer impact
  - Failure Points
  - Recovery procedures
Principles of Chaos

• Build a Hypothesis around Steady State Behavior
• Vary real world events
• Experiment in production
• Automate experiments to run continuously
• Minimize Blast Radius
Sample Chaos Experiments*

• Terminate random virtual server in a region.
• Subject entire fleet to high CPU/Memory within a region.
• Increase latency in one or more servers.
• Block access to a storage system.
• Failover a database to its secondary.
• Random killing of critical processes.

*For a generic three tier application, It is not comprehensive and depends on the application tooling support and maturity, but it is a good starting point.
Approach

Web Setup → Expected behavior
Sane Setup → Expected benchmarked behavior
Chaos Setup → Hypothesis (dis)proved behavior

Incorporate Learnings

Requests

For Comparing the results
Levels of Chaos

Sophistication

Presumed Resiliency

True Resiliency

No chaos Story

Frequent Crashes

Adoption
Why?

Image: www.salonuready.info
Chaos Engineering & DevOps
Chaos Engineering in CI CD Pipeline
Few tools to get started with Chaos Engineering

Simian Army:
- Chaos Monkey
- Chaos Gorilla
- Chaos Kong

- ChAP (Chaos Automation Platform)
Chaos Engineering as a part of DevOps Culture

Developers & stakeholders continually embrace failure as a way to prepare for and prevent it, resulting in stronger and more resilient applications.

Post the Chaos Experiments has been continuously validated in a CICD pipeline:

- Enable it in production, incrementally, in a random fashion (without notifying the support teams)
- Tooling must be mature enough to stop the experiment without further impacts.
- Initialize Chaos experiments introduced during random non-peak times, post maturity, move to peak times.
Industry Presence

- Amazon Web Services™
- Twilio
- Uber
- Microsoft Azure
- Kubernetes
- Facebook
- Netflix
- LinkedIn
- Dropbox
- Yahoo!
Industry Presence

Chaos Engineering
Companies, People, Tools and Practices
What?

Discipline of *experimenting* in a *distributed system* to *build confidence* in *system capability* in *turbulent conditions* in “PRODUCTION”.

https://principlesofchaos.org/
Everything fails, all the time.

- Werner Vogels -
VP & CTO at Amazon.com

The main point is no more to avoid failures, but to limit impact of those failures.
References & Resources

- https://principlesofchaos.org/
- https://medium.com/capital-one-tech/continuous-chaos-introducing-chaos-engineering-into-devops-practices-75757e1cca6d
- https://www.youtube.com/watch?v=6iMZqKdMMU
- https://www.youtube.com/watch?v=qHykK5pFRW4
Introduce Chaos, Increase Resilience

Thank you

Questions

Answers

Twitter: @GurtejPalSingh
LinkedIn: https://www.linkedin.com/in/gurtejpalsingh/