Chaos Engineering For Developers

Breaking Systems for Resilience

Conf42 Chaos Engineering 2024

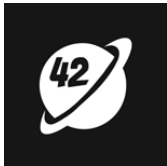
Hello, Dhiraj here

Software Engineer at Amazon Web Services(AWS)

Currently working in the multi-tenant, distributed, auto scale-out storage platform of AuroraDB which is purpose built for database engines.

Contributor/ Maintainer in OpenSearch Project

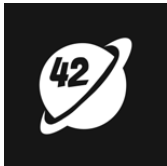
Contributed to various Open Source Projects revolving around K8s operators and helm charts. In OpenSearch Project, I maintain tools and softwares used to manage OpenSearch clusters in Kubernetes. I have bootstrapped charts and operators for OpenSearch on k8s.



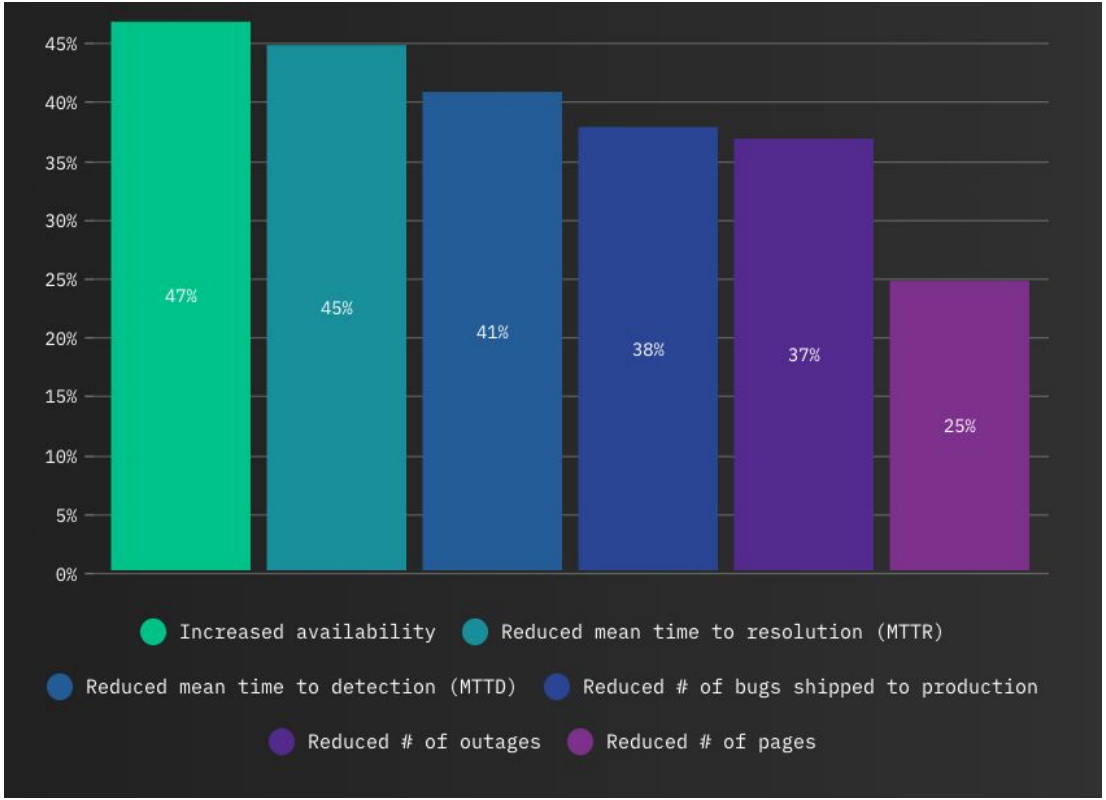
What is Chaos Engineering?

Practice of **intentionally introducing failures** into a system to identify weaknesses and improve its resilience.

By simulating real-world failures, chaos engineering allows organizations to **proactively identify and address potential issues** before they cause significant problems.



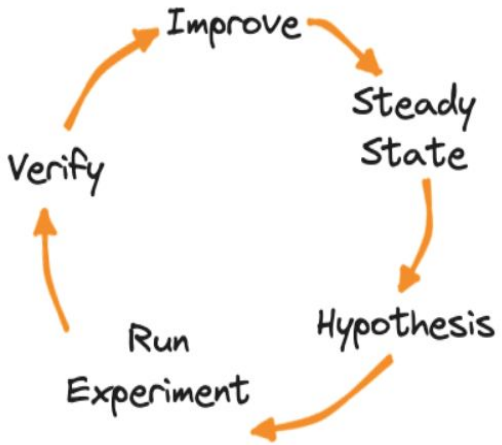
Why is it important for Developers?



Gremlin Survey Results for service metrics of companies after adopting **“Chaos Engineering as a Habit”**

What is the biggest inhibitor to adopting/expanding Chaos Engineering?

- **Lack of awareness** and experience.
- Followed closely by '**other priorities**'
- **>10% of engineers** (from *Gremlin yearly survey)* mentioned the fear that something might go wrong was also a prohibitor



What are different modes of Chaos Experimentation?

Manual Experiments

- Ad-hoc
- Gameday

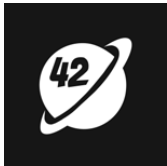
Automated Experiments

- CI/CD Pipeline
- Canary Release
- Continuous Experimentation

“For every dollar spent in failure, learn a dollar’s worth of lessons”

-Jesse Robbins

(Master of Disaster)



Few of the popular Open Source Tools

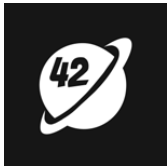
used for Chaos Engineering

Think Chaos while software development

1. Think about **external dependency failures** while designing

2. Write code assuming **sister systems will fail**

3. Inculcate **habit of “Chaos Testing”** as part of code testing



How to run controlled experiments?

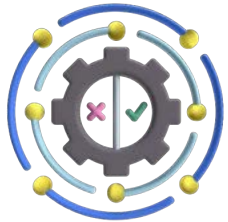
Step 1 : Identify boundary/scope of experiment

Step 2 : Build failure model for the service

Step 3 : Think through dependency failure

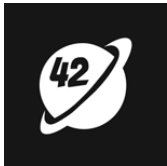
Step 4 : Inject Failures and monitor

Step 5 : Validate Results



Design a micro-service responsible for doing some CRUD operations and basic computations?

Let's think Chaos from a software developer perspective!



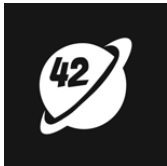
Best Practices in Chaos Engineering

Understand the **steady state** of the system

Build a resilience (or failure model) of your system to improve your hypothesis

Control blast radius of your experiments

Introduce **randomness in failure** injections



Best Practices in Chaos Engineering [Cont]

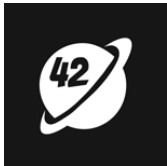
Test using **real world conditions**

It is a journey & does not start in production environments

Extensive **monitoring and logging**

Conduct **post-incident analysis** after each experiment

Start today, **experiment often**



Chaos Engineering Today

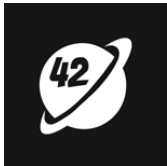
63% of 400+ IT professionals say they have performed chaos experiments

30% claim to run Chaos experiments in Production

Github has over 200+ Chaos Engineering related projects with 16k+ stars.

Teams who frequently run Chaos Engineering experiments are more likely to have >99.9% availability.

All major cloud providers like AWS, Azure etc. have their own managed service for doing Chaos experiments



Thanks

Let’s stay connected on X (Twitter)

@_TheAlgo_