What is chaos engineering, and is it safe to break production on purpose?

20 November 2026 · 4 min read · blogs.iictp.org/what-is-chaos-engineering

Chaos engineering is running controlled experiments that inject failures into a system, such as a lost server, a slow dependency or a network partition, to check that it behaves the way you believe it does. It is not random breakage: each experiment starts from a hypothesis and is designed to limit harm.

TL;DR The short version

- Chaos engineering runs controlled experiments that inject failures to check a system behaves as you believe it does.
- Each experiment starts from a hypothesis about steady state and is designed to limit harm, so it is not random breakage.
- Start with failures you already protect against, run in production only with good monitoring and a way to stop, and treat a failed hypothesis like an incident found early.

Read online, listen to it, or change the text size: https://blogs.iictp.org/what-is-chaos-engineering

How is an experiment structured?

The Principles of Chaos Engineering describe it: define steady state as a measurable output of normal behaviour, hypothesise that it will continue during the failure, introduce the failure, and look for a difference between the experimental group and a control group. A difference means you have found a weakness to fix.

Is it safe to run in production?

It can be, once you have practised elsewhere, have good monitoring and can stop the experiment at once. Keep the blast radius small, such as one instance or a small share of traffic, and expand only as confidence grows. Many teams start in staging and with scheduled game days, where people watch the experiment together.

What should you test first?

The failures you already expect and have built protections for: an instance dying behind a load balancer, a dependency timing out behind a circuit breaker, a failover to a replica. If those protections do not work, nothing more ambitious will.

What do you need before you start?

Observability good enough to see the steady state and any change in it, SLOs to tell you what matters to users, a runbook for stopping the experiment, and agreement from the people who own the system and will be on call.

What do you do with the results?

Treat a failed hypothesis like an incident found early: write it up, fix the weakness, and re-run the experiment to confirm the fix. Experiments that pass can be automated and run regularly, so protections keep working as the system changes.

Related credential: Certified Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Principles of Chaos Engineering. principlesofchaos.org.
2. Basiri, A. et al. Chaos Engineering. IEEE Software, 33(3), 2016.
3. Rosenthal, C. and Jones, N. Chaos Engineering: System Resiliency in Practice. O'Reilly, 2020.

Published by IICTP, an independent technical credential body, on 20 November 2026. The current version of this post is at https://blogs.iictp.org/what-is-chaos-engineering. More guides: blogs.iictp.org.