INTRODUCTION

Chaos testing involves carefully introducing faults or breakdowns into your infrastructure in order to assess how well the system will function in the event of a disaster. This is a useful technique for practicing, preparing for, and minimizing downtime and outages before they happen. It is also a way to evaluate the reliability of a system by anticipatorily simulating and spotting errors in a specific environment before they result in unplanned downtime or a bad user experience.

Chaos engineering is made up of five main principles:

1. Ensure your system works and define a steady state

   To accomplish this, you must define a "steady state" or control as an objectively measured system output that denotes typical operational behavior .

2. Hypothesize the system's steady state will hold

   It is necessary to assume that a steady state will persist under both control and experimental circumstances once it has been identified.

3. Ensure minimal impact to your users

   It's necessary to actively strive to break or disrupt the system during chaos testing in order to reduce the blast radius and any adverse effects on your users. Your team will be in charge of making sure that each test is concentrated on a particular area, and you should be prepared to respond to incidents as needed.

4. Introduce chaos

   You can begin running your chaos testing applications once you are certain that your system is operational, your team is ready, and the blast radius has been contained. A server crash, broken hardware, severed network connections, and other real-world occurrences should all be simulated using various variables that you introduce. To observe how your service or application would respond to these events without directly affecting the live version and active users, it is best to test in a production environment.

5. Monitor and repeat

   The secret to chaos engineering is to test frequently while adding disorder to identify any flaws in your system. Chaos engineering aims to refute your second hypothesis while simultaneously creating a stronger, more reliable system.