Muhammad Ahmad	19L-2300
Usama Iqbal	19L-2232
Zaviyaar Bin Irfan	19L-2225

System Reliability Testing Using Chaos Engineering

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.

HOW DOES CHAOS TESTING WORKS?

In chaos engineering, each experiment begins with the introduction of a particular flaw into the system. The administrators compare what actually occurred to their predictions later on.

Here's the step-by-step flow of chaos engineering experiments in practice:

1. The steady state of the system is defined as a measured output that represents typical behavior.
2. Suggest on the output of the system and how it will differ between the experimental and control groups.
3. Introduce variables that represent actual occurrences, such as traffic spikes, hardware and software failures, and non-failure events.
4. By contrasting the steady state of the system in the control and experimental groups, work on refuting the theory.

Two groups of engineers typically participate in chaos engineering experiments. The first group typically manages the failed injection, while the second group takes care of the outcomes.

An example of testing tool : In order to address the demand for continuous and consistent testing, Netflix started chaos testing their system throughout their migration to Amazon web services. To this end, they built various "chaos monkeys." These chaotic monkeys were introduced into a system to imitate various real-world conditions and introduce unique faults, such as network latency, instances, missing data segments, etc.

Each chaos monkey had its own name and job, including:

- Latency Monkey: Induces artificial delays
- Conformity and Security Monkeys: Hunt and kill instances that don't adhere to best practices
- Janitor Monkey: Cleans up and removes unused resources
- Chaos Gorilla: Simulates an entire Amazon availability zone outage

Collectively, these and more chaos monkeys are now known as the Simian Army.

PROS

- IT and DevOps teams are better able to recognise and address problems that other testing methods might miss.
- Because of proactive and continuous testing, unplanned downtime and outages are far less likely to happen.
- improves the system's integrity
- Great for scaling up to huge, complicated systems

CONS

- Desktop software or smaller systems
- Services and programmes that are not essential to the operation of the business
- Application settings without customer service level agreements requiring constant uptime 24 hours a day
- Systems where errors are tolerated as long as they are fixed at the end of the day

TOOLS USED FOR CHAOS TESTING

1. **Chaos Mesh**

   An open-source cloud-native tool is called Chaos Mesh. Chaos Mesh assists businesses in identifying system irregularities that might happen during various stages of the development, testing, and production processes by using a variety of fault simulations.

   Attacks that test network latency, system time manipulation, resource use, and other factors can be launched using Chaos Mesh. Various types of experiments can be modified and managed within predetermined time frames using the Chaos Dashboard.

2. **Chaos Monkey**

   A tool for chaos engineering that was first developed by Netflix developers is called Chaos Monkey. It was created to aid in testing the system's resilience and dependability after a move to the AWS cloud. By implementing ongoing unpredictable attacks, the software operates. Chaos Monkey employs the core strategy of terminating one or more instances of virtual machines.

   Chaos Monkey's flexibility enables simple scheduling and careful supervision. Although the technique is simple to reproduce, problems may arise if users are not ready for the fallout from

attacks. Prior to deployment, users can check for outages, but they must be able to create and modify custom Go code.

3. **Gremlin**

   The first hosted chaos engineering platform, Gremlin, was created to increase web-based dependability. Gremlin, which is available as software-as-a-service (SaaS), can assess system resilience using a variety of attack patterns. Users input data into the system to determine the kind of attack that will yield the best outcomes. Comprehensive infrastructure assessments can be facilitated by performing tests in tandem with one another.

CHAOS TESTING in DevOps

A DevOps framework works well with chaos engineering. Typically, a DevOps engineer like the XA(Experience Assurance Professional) is responsible for chaos engineering. This individual is in charge of creating the various testing situations, carrying out the tests, and monitoring the outcomes. They are also in charge of making sure the consumer is affected as little as possible.

The DevOps engineer must walk a very narrow line while testing. One method is trying to make the system crash while adding chaos to evaluate the system's integrity (hence, why this is best done in a production environment). On the other hand, running haphazard or careless tests can potentially result in a system crash and degrade user experience.

REFERENCES:

1. https://www.cigniti.com/blog/guide-chaos-engineering/
2. https://www.pagerduty.com/resources/learn/what-is-chaos-testing/
3. https://www.blameless.com/blog/chaos-engineering