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.