CECS 574 Fall 2023

Chaos Engineering on Kubernetes Summary

Bryan Linares, Rutvi Patel

DevOps teams who manage the infrastructure of the company are often worried about services going down/breaking and not knowing about it. Using the Kubernetes Cluster Orchestration platform to deploy services and applications has become popular due to its scalability and efficiency. Kubernetes is a platform designed for managing distributed systems and is powerful when working with large data, large traffic, and AI Applications. However, every modern technology comes with many advantages but also some risks. Some of the risks that DevOps accept when using the Kubernetes clusters include Network loss, High CPU Usage, Bad Configurations, Overloading, and all the stability-threatening issues one can think of. The concept of Chaos Engineering allows us to counter these blind spots and helps ensure the resiliency of the distributed system environment. Chaos Engineering is a methodology and framework for introducing controlled failure and faulty scenarios to a distributed software system to test and verify its resilience. Chaos Engineering in general consists of 4 phases: the Hypothesis, Designing Experiment, Analyzing Results, and Fixes. Tools like Chaos Toolkit and Litmus allow you to test all the phases of Chaos Engineering in a controlled, programmable, reusable, and detailed Environment. The practice was popularized prominently by Netflix and their Chaos Monkey tool which allowed them to simulate outages in servers.

The open-source command line Chaos Toolkit (chaostoolkit.org) provides an extensible and simple to use set of tools to inject faults and specify rollbacks into a system. It has an additional driver to integrate with container orchestration and infrastructure systems like Kubernetes to use in the command line. It primarily uses a textual base for defining functions and values to measure and take action with. For the toolkit, within these files the engineer creates a set of probes and actions that the toolkit reads to measure variables and cause disruptions in the system. The hypothesis can be any number of measurable values, in our example checking at HTTP code is used, but DNS, TCP, and other available values in the respective environment can be checked. The toolkit requires the definition of a steady state, an uses the probes to collect data on metrics like response timing error rates, and node utilization. The actions are the fault injection methods that can be simulated, like adding latency, killing processes, or draining nodes and other outages. There are many actions defined in the toolkit libraries and the drivers that extend them to the platforms the user needs. You can use JSON or YAML files and output can be observed with similar openly compatible data output. Chaos Toolkit is completely free and open source.

Litmus is a comprehensive platform designed for Chaos Engineering, offering tools and services to help organizations actively identify weaknesses in their software systems. At its core, Litmus enables chaos experiments that simulate real-world failures, allowing teams to assess the resilience of their applications. The platform provides several key services, including ChaosHub, an extensive repository of pre-built chaos experiments and workflows that can be readily deployed. Currently, there are 58 different prebuilt experiments available to test on your environment. Additionally, Litmus offers an Environment service that helps users define and manage targeted chaos environments, ensuring that experiments are conducted in controlled settings. One can have multiple environments as well as multiple infrastructures within an environment. The Experiments service allows for the creation, scheduling, and monitoring of chaos tests, and it provides valuable insights into system behavior during failures. Good experiment examples include CPU Hog, Delete pod, Network Loss, Network Latency, Filled Disk Space, Bad Configurations etc. Finally, the Probe service helps collect and analyze metrics and other data during experiments, facilitating a deeper understanding of system responses to chaos, thereby aiding in the improvement of system reliability and robustness. Some of the commonly used probes are Grafana and Prometheus. Litmus empowers organizations to enhance their software infrastructure's resilience and fault tolerance through systematic chaos engineering practices.

In recent times, the practice of Chaos Engineering can be commonly found to be combined into the workload of DevOps and the job title of Site Reliability Engineer. By incorporating Chaos Engineering into their work, they are better equipped to identify vulnerabilities and weaknesses in their systems, leading to more resilient and robust applications by using a standard scientific methodology. Moreover, the Cloud Native Computing Foundation (CNCF), a prominent organization in the cloud-native ecosystem, has recognized the significance of Chaos Engineering. CNCF has actively embraced this practice, acknowledging its role in enhancing the reliability and performance of cloud-native applications. Tools are widely used and vary depending on the organization that made and needs them.

In conclusion, Chaos engineering is like performing a fire drill in a high-rise building. Just as a fire drill is a controlled simulation of a real emergency, chaos engineering involves creating controlled scenarios of system failures in a software environment. All phases of Chaos Engineering and the results provide an important insight into the Kubernetes environment. This practice has gained value in the DevOps and Site Reliability Engineering (SRE) fields to strengthening the core resilience of distributed systems.