What is Chaos Engineering?

Description

Use Case

Just Consider for reality ðŸ˜Ž you have a website or a service which is worth millions and billions of dollar and all of a sudden it goes off for a significant amount of time. You are totally unaware of what impact or loss this will incur. Till date you were so confident that you have used the best possible technologies out there and experienced developers/architects worked on it. But now you are under the radar of the management and the customers ðŸ¤•.

I have personally faced this situation while I was working for a renowned bank. The fact of the matter is we all are afraid of this situation and always want to avoid any service outage or stoppage in production. So we should all be ready for such situation by simulating such scenarios every now and then.

Introduction to Chaos Engineering

Chaos engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production. Chaos engineering is a discipline that directly addresses System Availability and affects your culture. But we should always strive and ensure nothing breaks in production. Even if it breaks then we should see how will our application behave or handle. Along with our applications, the platforms and infrastructure also should help in tackling this situation. Chaos engineering is nothing but experimenting something in production which no one can expect to happen. So that we can identify how the application or system will behave when this situation will arise. It will enable us to find the loopholes in the system and fix it accordingly.

It is something similar to the Mock Fire Drills we perform in our office's. Hope this was close to what I am trying to explain. ðŸ™†â€â™,ï¸• Let me know in the comments.

Principles

- **Build a Hypothesis around Steady State Behavior**

Consider we have a banking application, and we are assuming that our application is in a steady state behavior, it is working absolutely fine. Some day if there is a sudden outage while a customer is doing a money transfer from one account to another, how will the system behave? will the money get deducted from the customer account and not deposit into the destination account? where will the funds go?

This is one of the use case that can happen. In that way we need to list all possible negative scenarios

and our goal should be to work on handling or fixing those missing pieces.

- **Vary Real-world Events**

Chaos variables reflect real-world events. Here we need to think of real world events like sudden increase in traffic or server damage or hardware failure or any non expected real world events which will disrupts the steady state of the application.

- **Run Experiments in Production**

Systems behave differently depending on environment and traffic. Application feature may work fine in Development environment , but may timeout or respond slowly. You have to run your experiments in real production environment. Unless you run in production, you wont be able to know the impact.

- **Automate Experiments to Run Continuously**

Chaos Engineering builds automation into the system to drive both orchestration and analysis. The experiment has to be carried out from every now and then to see if any new issue surfaces. Experiments can be either monthly or quarterly basis.

- **Minimize Blast Radius**

Identifying the failure and reducing the impact of the failure without causing much damage to the live customers. Experimenting in production has the potential to cause unnecessary customer pain. While there must be an allowance for some short-term negative impact, it is the responsibility and obligation of the Chaos Engineer to ensure the fallout from experiments are minimized and contained.

Tools Available

Netfix has a Simian Army. It is nothing but a group of tools used to test chaos situation in production every now and then. Below are the tools by Netflix

Latency Monkey will purposefully delay the requests and see what happens to the requests.
Chaos Money , It goes and randomly kills a microservice and see what happens to the flow or behavior.
Chaos Gorilla, it kills the entire availability zone. Availability zone in case of AWS can be NA-West .
Chaos Kong is another tool which will kill the entire region randomly , which can be NA-East or NA-West to check the behavior of the whole system.

Similar to Netflix every large organization has their own strategy to handle these situations. Facebook has something called as Facebook Storm.

I hope you liked the post ðŸ˜‰ Keep visiting and learn new things.

Category

1. AWS
2. Cloud
3. Design

Date Created
December 23, 2021
Author
kk-ravi144gmail-com