What is Chaos Engineering?

Principles of Chaos Engineering defines Chaos engineering as “the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production”.

http://principlesofchaos.org

Benefits

Benefits of Chaos Engineering

- Promote Innovation
- Elevate Partnership
- Improve Incident Response
- Generate Knowledge
- Increase Reliability and Resiliency

Chaos in NAM

Chaos Backlog

- Top incidents
- New components being deployed
- Big changes in design
- By request

Chaos Testing

- **Timing :** Monday and Tuesday following the monthly release weekend
- **What Will be done :** SRE team and/or PERF team will identify and execute some planned scenarios
- **Tools** : Depend on the application (Manual, Gremlin, ApeArmy, etc)

Game Day

- **Timing :** Monday and Tuesday following the monthly release weekend
- **What Will be done :** SRE team would be initiating a Production Stress Testing of NAM CORE
- **Tools** : AMP or Starfleet

Wheels of Misfortune

- **Timing :** Once every quarter
- **What Will be done :** SRE team along with stakeholders would be role-playing scenarios to test techniques for responding to an emergency
- **Tools** : Assessment form

How to introduce Chaos

Gremlin

It’s a chaos engineering platform which is available as a SaaS solution, that allow to inject failures at various layers of the system. It can assess system robustness using one of three attack types.

Chaos Monkey

It’s a tool created by Netflix to test resilience of their applications. It simulate failures by randomly terminating instances within the system. Use Spinnaker and MySQL and setup a cron-job to run Chaos Monkey daily schedule.

Ape Army

In House tool, Chaos Engineering platform. The overall approach is to bring down random services/pods/network fault/latency using rest APIs or management extensions while the load is being injected to the application under test. Once the services/pods come up, the validation will be done to ensure there is no loss of messages/trades which were placed initially.

Manual

Basically consists in manual manipulation of the environment, like updates in configuration parameters, disabling components, service restart, etc.

Gremlin

Overview

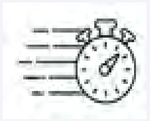
Experiments in Performance Environments

Replicate Prod Issues and planned failures during tests

Assess application flow, devise Steady State behavior and observe hypothesis

Analyze the results and learnings and feed it back

New Services / Enhanced Services



Increasing Velocity of Releases

Improving Brand Trust

Reduce Downtime

Business Goals

Slow / unavailable dependencies

Validate Resiliency Patterns

Peak season readiness

Accelerate Transformation

Engineering Goals

Past Incident History

MTTD / MTTR

Alerting & Monitoring observability

Incidents & Failures

Scope of the Gremlin Exercise

Methodology

1. Steady State Assessment
2. Hypothesis & Experiments
3. Execute Experiments “Test Days”
4. Analyze Results

Fault Injection Attacks

Resource Attack
- ✓ High CPU,
- ✓ High Memory
- ✓ High I/O Load
- ✓ VM, Host & Disk Failures (SAC Approval Pending)

State Attack
- ✓ Shutdown
- ✓ Process killer
- ✓ Time Travel

Network Attack
- ✓ Latency & Packet Loss
- ✓ Blackhole Network Connectivity Failure

Performance Degradation

Service Degradation

Network Degradation

Blast Radius
- ✓ Application Dependencies
- ✓ Stimulating Maximum Capacity
- ✓ Point Break tests

What is Wheels Of Misfortune?

A Wheel of Misfortune is a role-playing scenario to test techniques for responding to an emergency. The purpose of this exercise is to learn through a purely simulated emergency, using a traditional role-playing setup, where engineers walk through the steps of debugging and troubleshooting. It provides a risk-free environment, where the actions of the engineers will have no effects in production, so that the learning process can be reinforced through low-stakes struggles.

Benefits of this Exercise:

- Encouraging reverse engineering, statistical thinking, and working from fundamental principles
- Shrinking the mean time to mitigate (MTTM) of production incidents
- Creating contained but realistic breakages for engineers to fix using real monitoring, documentation and tooling
- Practicing elevated stress levels during outages and methods to control those situations
- Role-playing theoretical disasters as a group, to intermingle a team's problem-solving approaches
- Creating Stellar Reverse Engineers and Improvisational Thinkers
- Disaster role playing; Emphasize and promote blameless culture and Share the experience of past incidents

How to Execute WOM

- The GM is an experienced engineer who knows how the services work and interact to respond to the operations requested by the player(s)
- GM, runs a scenario where some non-player characters get into a situation and interact with the players, who are the people playing the role of the emergency responders.
- Simulation of MIM outages that recently occurred or Chaos Engineering component testing of an actual outage

Roles

GameMaster (GM)

Volunteer

Non-Player Characters

Steps in the Role Play

GM describes how the player(s) becomes aware of the service breakage: an alert, IM from ICM/Contact Center /Manager

Player then gives the clear explanation of what they want to do, dashboards they want to visualize, diagnostic commands to run, view playbooks, etc.

GM would evaluate the players knowledge of the systems and their troubleshooting capacity along with incident management process, declaring a major outage and incident communication

The rest of the team should be spectators, unless specifically called in by the GM or the player. If the scenario is taking too long, or the player is stuck on one part, allow suggestions from the audience, or provide hints.

Finally, once the scenario is concluded GM should state clearly that the situation is fixed. Allow some time at the end for debriefing and discussion, explaining the background story that led to the emergency & indicating the contributors to the situation.