



# Chaos Engineering

TASO RODRÍGUEZ MEANA Y RAÚL LÓPEZ CALVO

# Índice

- PRINCIPIOS DE LA INGENIERÍA DEL CAOS
- VENTAJAS/DESVENTAJAS
- COMO REALIZAR INGENIERÍA DEL CAOS
- NETFLIX
- CONCLUSIONES

# ¿Problemas de los sistemas distribuidos?

- ▶ Los sistemas distribuidos de hoy en día son muy complejos.
- ▶ La gran complejidad es inherentemente caótica
- ▶ El caos genera errores impredecibles



# ¿Qué es la ingeniería del caos?

- ▶ Los sistemas de hoy en día son muy complejos.
- ▶ Fallos en el sistema causan pérdidas.
- ▶ Capacidad de mejorar la resistencia.



# Ventajas

- ▶ Pruebas de vulnerabilidad más completas
- ▶ Mitiga riesgos, aumenta confianza y reduce carga de trabajo
- ▶ Menor coste de mantenimiento
- ▶ Para los clientes mayor disponibilidad

# Desventajas

- ▶ No existe una principal desventaja, solo dificultades
- ▶ Elegir la métrica puede ser difícil según el contexto.
- ▶ Pruebas sobre el sistema en producción, hay que controlar el daño para no producir pérdidas
- ▶ La ingeniería del caos “no debería de existir”

# Ingeniería del caos, implementación

4 pasos fundamentales:

- ▶ Establecer un estado de comportamiento normal, métrica
- ▶ Este estado continuara en el grupo de control y el experimental
- ▶ Introducir variables del mundo real, fallos de servidores, de red, aumento de trafico...
- ▶ Buscar diferencias entre el estado en el grupo de control y el experimental, una vez encontradas vulnerabilidades mejorar el sistema para que sea capaz de responder.

# Ejemplo real: Netflix

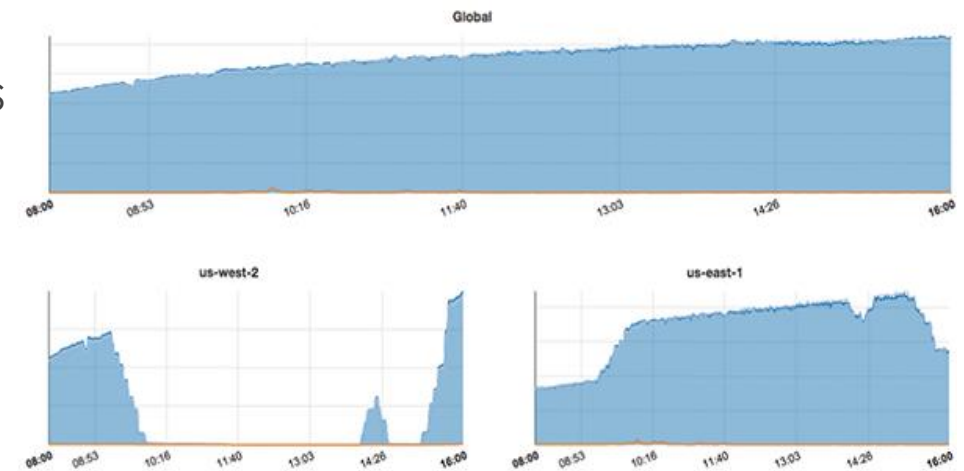
- ▶ Aplicación que da servicio a millones de personas todos los días
- ▶ Chaos Monkey: servicio que mata un server de la rama de producción de ASW, en el momento parecía una idea muy rara.
- ▶ La idea funcionó, estos experimentos permitían mostrar problemas en los sistemas en los que se probaban
- ▶ El equipo de Netflix fue el que le dio el nombre: Ingeniería del Caos





# Netflix, avanzando en la fórmula

- ▶ Tras el éxito de Chaos Monkey deciden continuar con la fórmula
- ▶ Chaos Kong: mata no un servidor sino una región entera de AWS
  - ▶ Situación rara, pero con precedentes (20-sep-2015)
  - ▶ Objetivo: que los usuarios no noten la caída.
- ▶ Ejemplo ejecución, King Kong tumba el servidor de US-west, el tráfico se redirige al servidor US-east.
- ▶ En la gráfica se puede ver el numero de reproducciones



# Conclusiones

- ▶ Sistemas muy complejos inherentemente caóticos
- ▶ Resistir los errores vs prevenirlos
- ▶ Enfoque proactivo vs reactivo
- ▶ Permite proveer servicios mas resilientes
- ▶ Provee de práctica a equipos de desarrollo antes situaciones de fallo
- ▶ Abarata costes y mejora el proceso de desarrollo

# Bibliografía

- ▶ <https://netflixtechblog.com/chaos-engineering-upgraded-878d341f15fa>
- ▶ <https://www.infoq.com/articles/chaos-engineering-security-networking/>
- ▶ <https://principlesofchaos.org/>
- ▶ [https://en.wikipedia.org/wiki/Chaos\\_engineering](https://en.wikipedia.org/wiki/Chaos_engineering)