

Chaos Engineering en AWS: Resiliencia más allá de la confianza en la nube

Ramon Esparza



Agradecimientos a nuestros patrocinadores



¡ Soy Ramón Esparza



🎓 Ingeniero en
Mecatrónica

🛡️ Staff SRE en
Wizeline



*** ¿Qué tan seguro estás de que tu aplicación en AWS sobreviviría si mañana falla una lambda?**

*** ¿Qué prefieres: descubrir los puntos débiles de tu sistema en horario laboral... o a las 3 a.m. en producción?**

*** ¿Conoces al 100% el sistema o tu aplicación?**



Introducción: ¿Qué es Chaos Engineering?

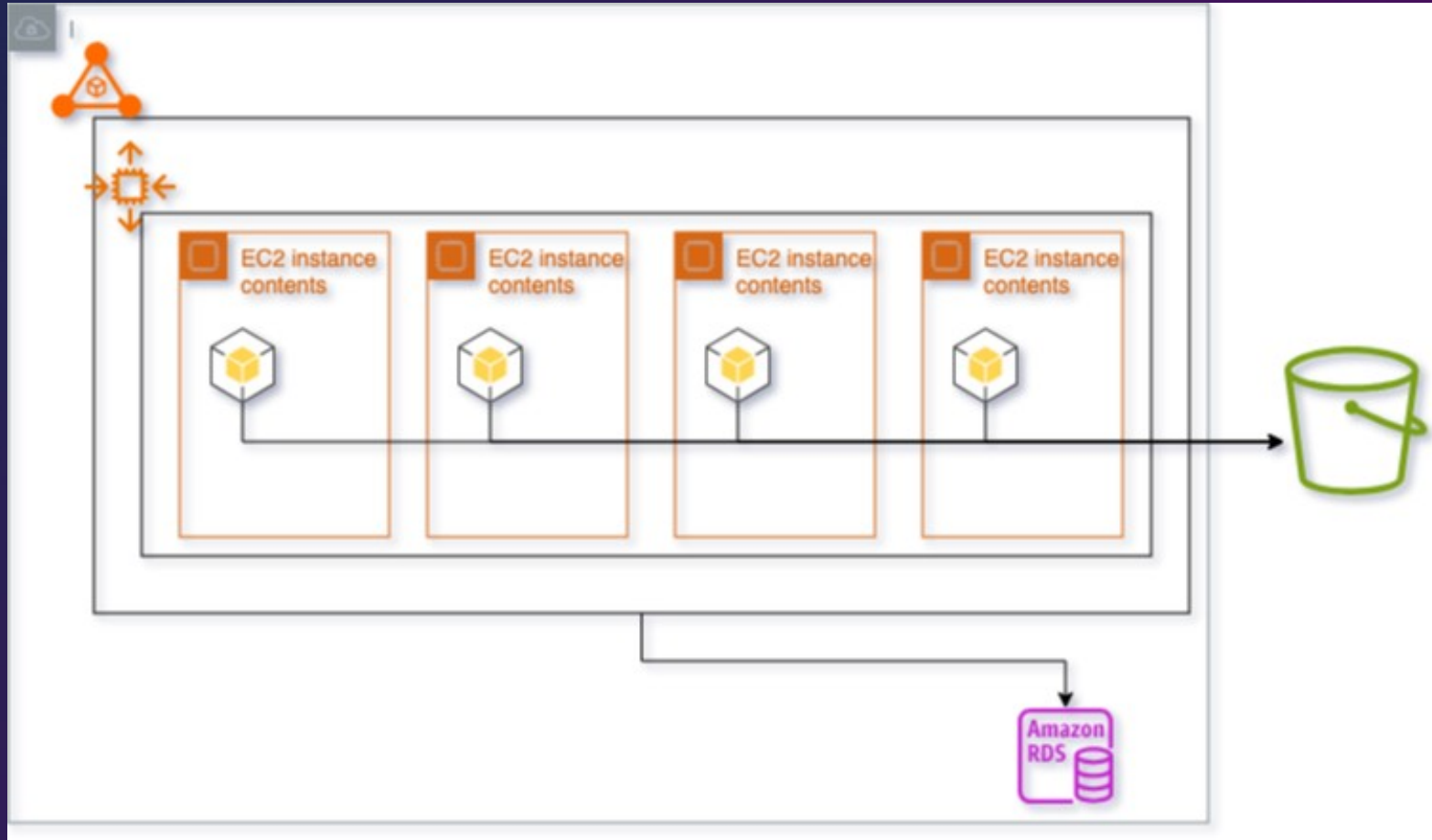
* Destruir para aprender.



Complejidad: Desde el servidor hasta los microservicios.



Complejidad: Desde el servidor hasta los microservicios.



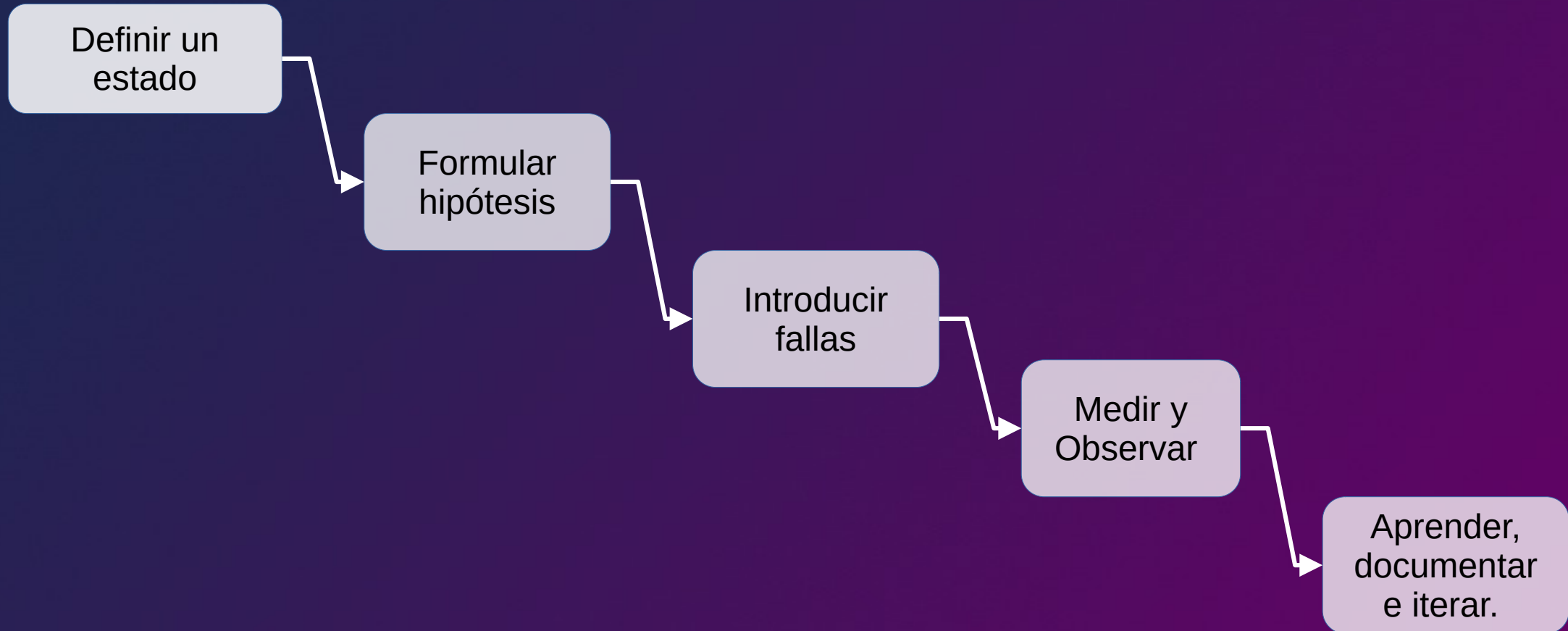
¿Por qué romper nuestros sistemas?

- Chaos Engineering no se trata solo de romper cosas, sino de hacerlo de forma **controlada, medible y repetible**, mediante la **experimentación**.
- Las fallas ya no son simples ni aisladas. Romper de forma controlada nos ayuda a entender cómo reaccionan esos componentes bajo presión.
- La organización permite aprender, documentar y evolucionar los sistemas.





El ciclo de vida de los experimentos.



Organiza los experimentos.

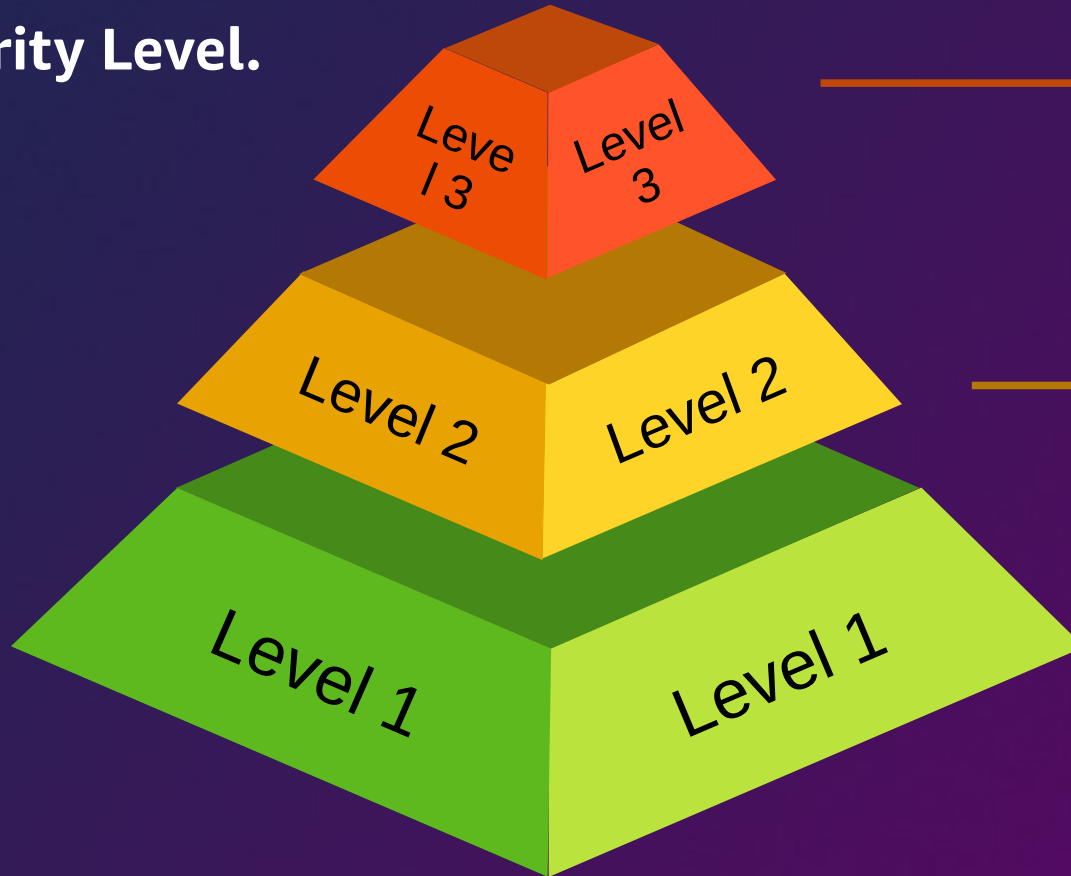
- **Chaos as Code:** Genera plantillas como código, versionar en GIT.
- **Automatiza la ejecución:** Crear flujos en Github, AWS FIS, Jenkins.
- **Etiqueta los recursos que quieres afectar.**
- **Observabilidad y Alarmas.**



Escalar la experimentación.

➤ **Chaos Playbook.** Un manual sobre las prácticas que se han ejecutado hasta el momento.

➤ **Chaos Maturity Level.**



Experimentación Continua

Experimentos automatizados y ejecución constante.

Chaos days.

Periodos de tiempo donde se concentre en el caos y resiliencia.

Ad-hoc

Experimentos simples y sencillos.





¡Muchas gracias por tu atención!

Nos vemos en 2026 



Con el apoyo de :

