SELA|DEVELOPER|PRACTICE
December 12-16, 2021

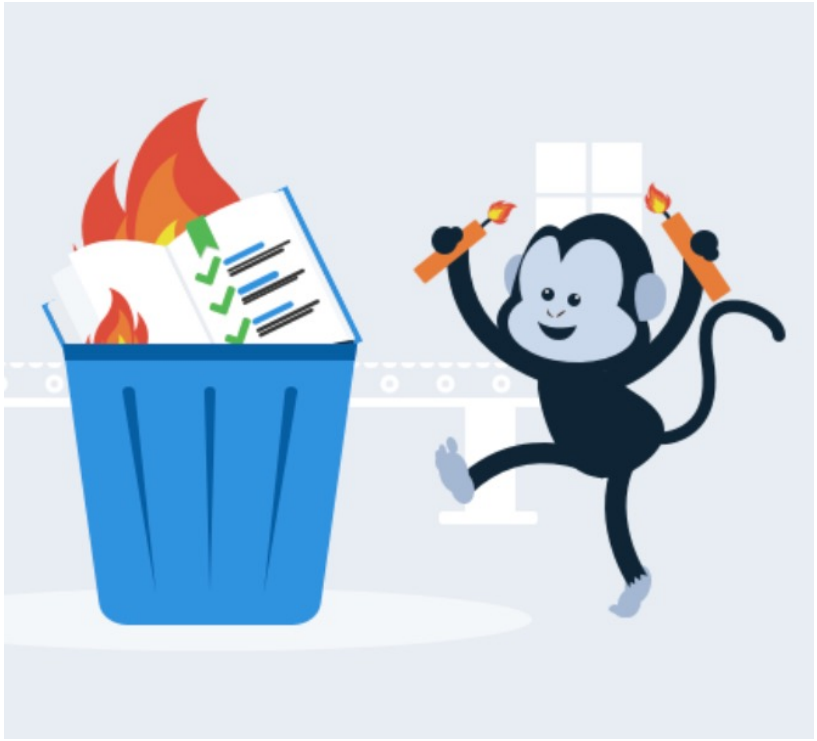
Playing with Chaos Engineering

Speakers: Leon Jalfon & Jonathan Jalfon

December 15, 2021

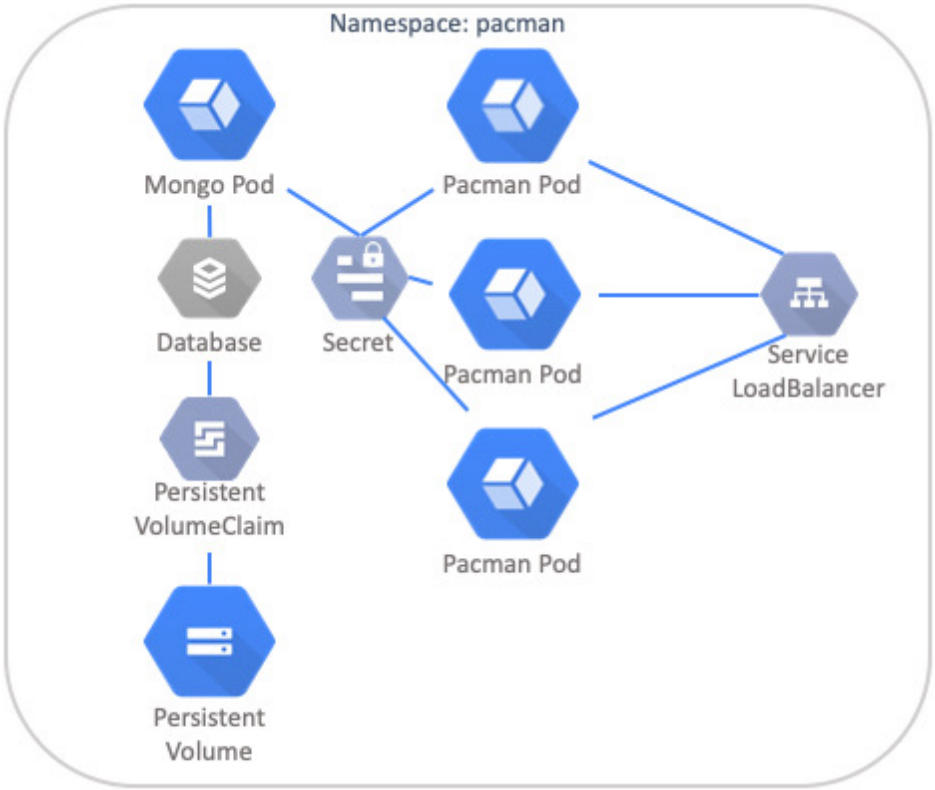
Agenda

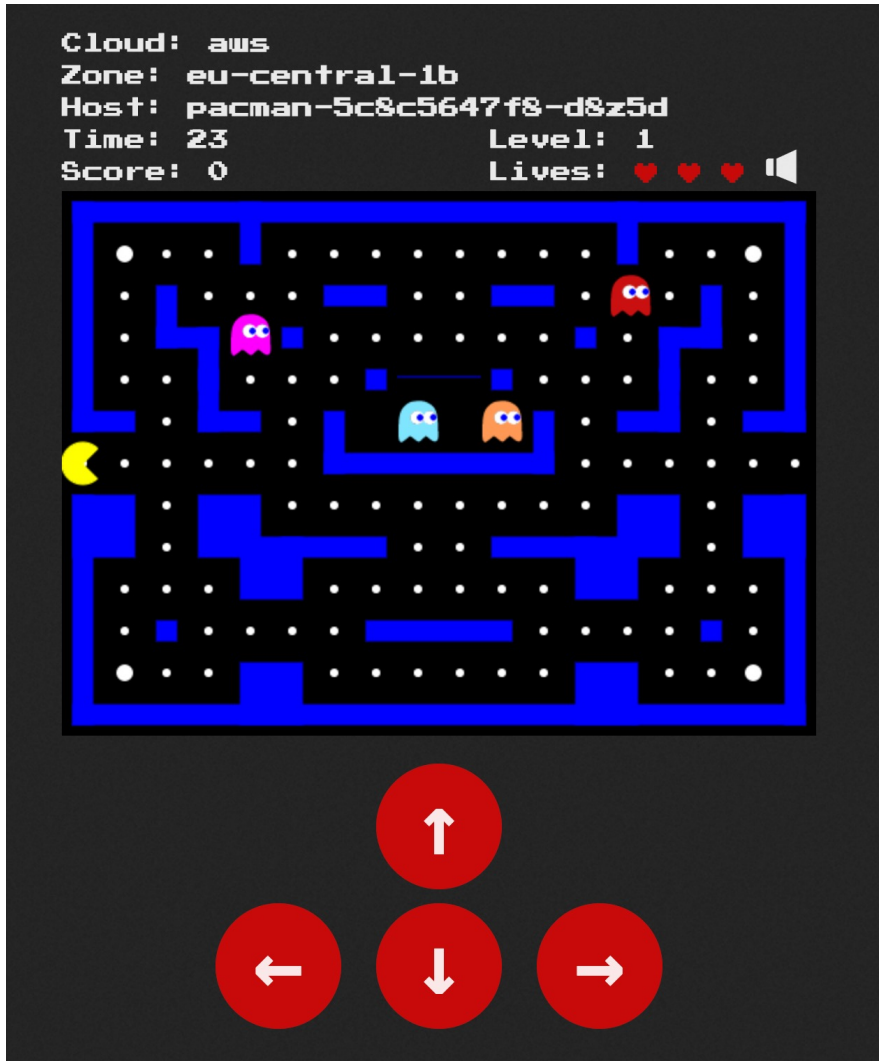
- Meeting our Demo Application
- Introduction to Chaos Engineering
- Chaos Engineering on AWS
- Summary



Demo Application

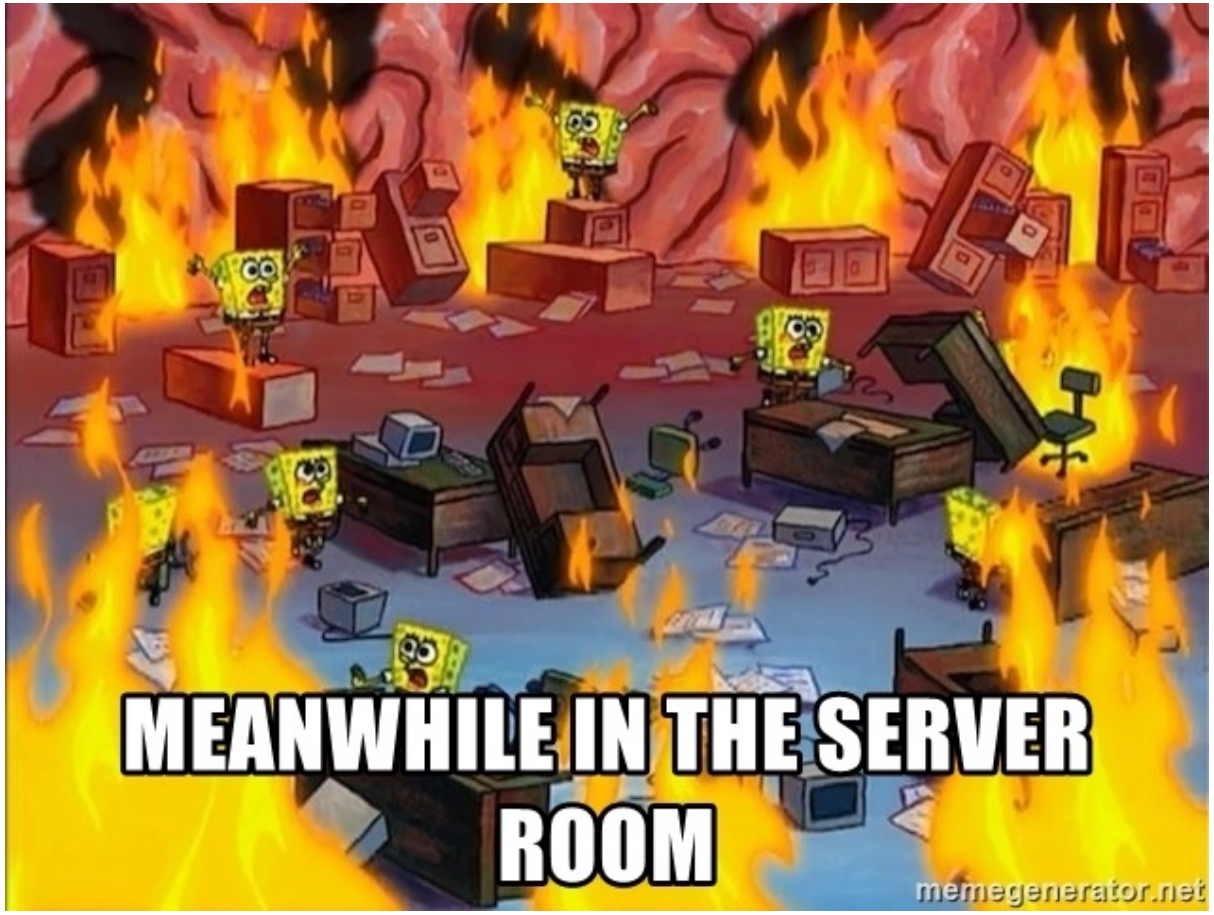
https://pacman.seladevops.com





Eventually, Systems WILL FAIL

- Single Point of Failure
- Unreliable Network
- Slow Processes
- Unexpected Load
- Cascading Failures
- Cloud Outages



Demo 01: Testing Resiliency

Please access our demo application and play around while we cause some chaos to test the application resilience

https://pacman.seladevops.com

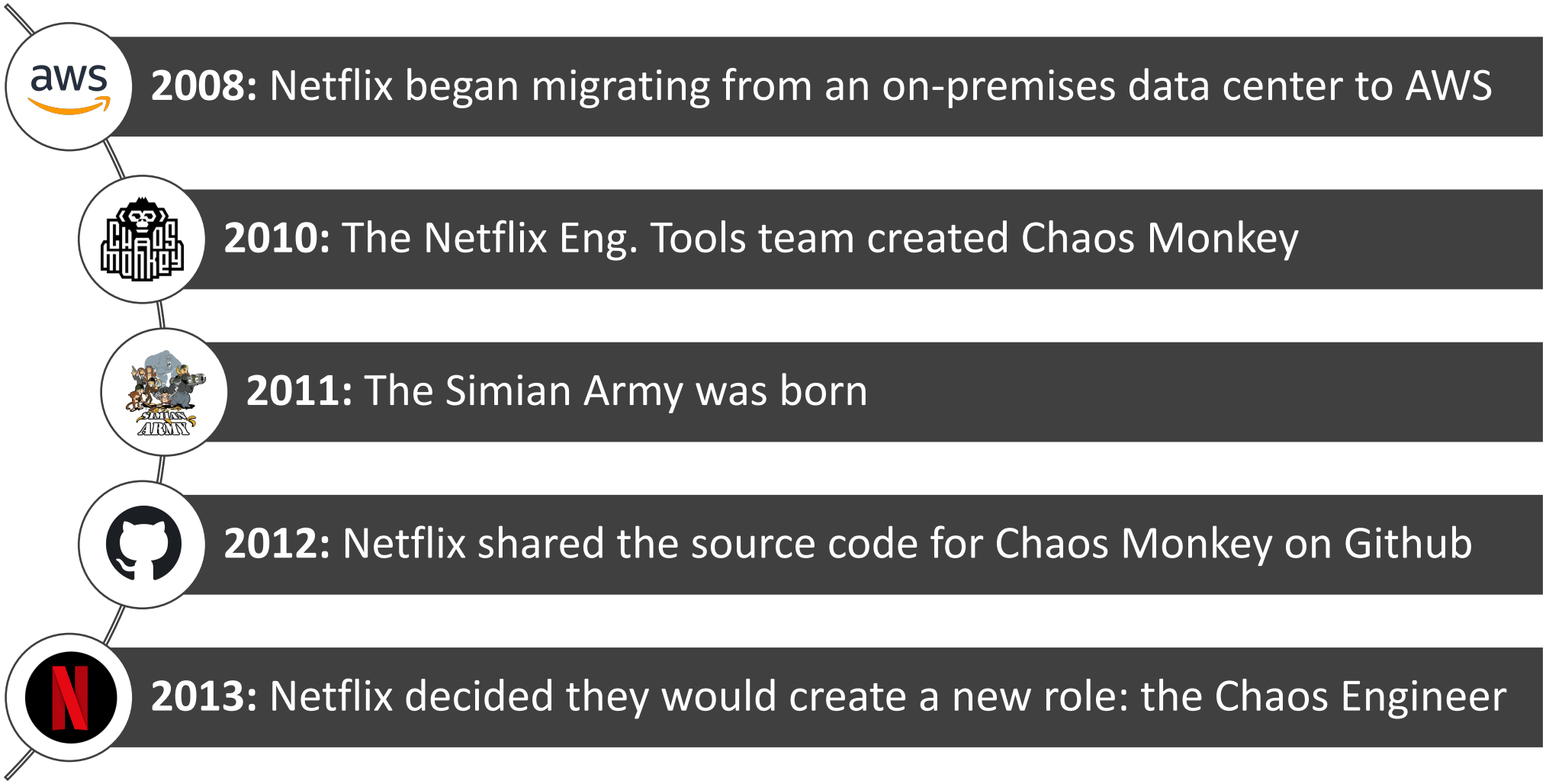
Defining Chaos Engineering

"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"

Principles of chaos: https://principlesofchaos.org/

Origin of Chaos Engineering

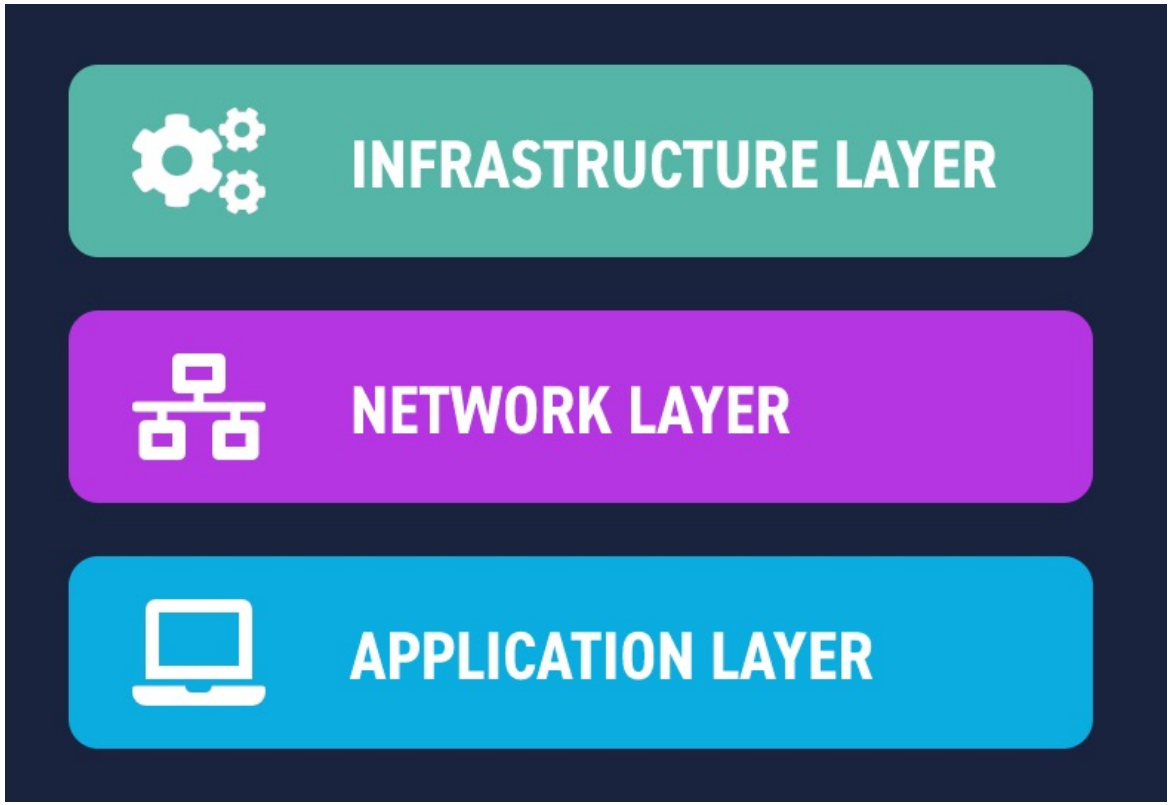
Origin of Chaos Engineering



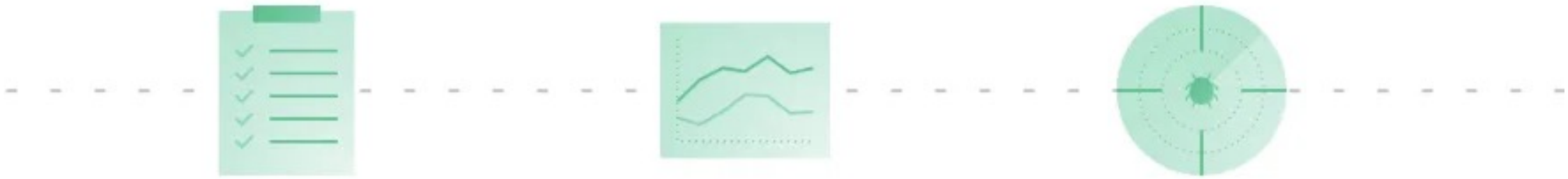
Netflix - Chaos Monkey & Simian Army

Injecting Chaos

You can inject chaos at any layer to increase system resilience and system knowledge



Phases of Chaos Enginieering



I. Plan an Experiment

Create a hypothesis. What could go wrong?

II. Contain the Blast Radius

Execute the smallest test that will teach you something.

III. Scale or squash

Find an issue? Job well done. Otherwise increase the blast radius until you're at full scale.

Choose your Chaos Engineering Tool

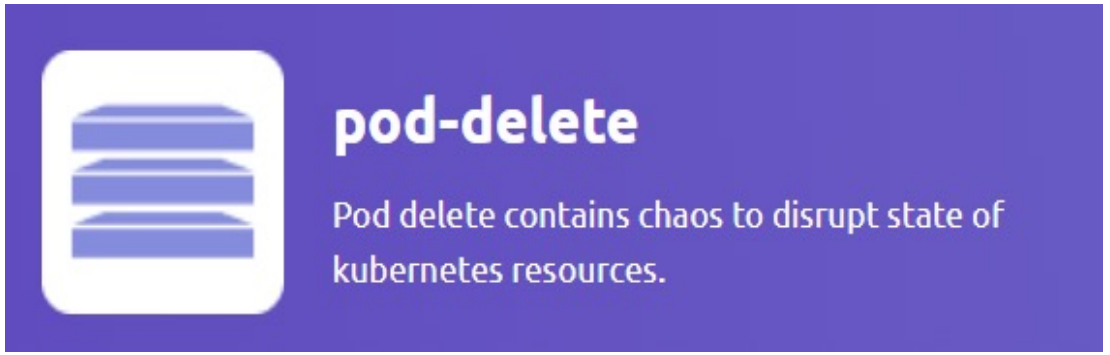
	Works with	Total attack types	Application attacks	Host attacks	Container/Pod attacks	GUI	CLI	REST API	Metrics/Reporting	Attack Sharing	Attack Halting	Attack Scheduling	Target Randomization	Custom Attacks	Health Checks
Gremlin	Containers, Kubernetes, bare metal, cloud	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Chaos Monkey	Amazon Web Services (requires Spinnaker)	1		✓		✓			✓			✓	✓		✓
ChaosBlade	Containers, Kubernetes, bare metal, cloud	40	✓	✓	✓		✓	✓	✓		✓				✓
ChaosMesh	Kubernetes	17		✓	✓	✓	✓		✓		✓	✓	✓		✓
Litmus	Kubernetes	39		✓	✓	✓	✓		✓	✓	✓	✓	✓		✓
Chaos Toolkit	Containers, Kubernetes, bare metal, cloud	Based on driver		✓	✓		✓			✓			✓	✓	✓
PowerfulSeal	Kubernetes	5+		✓	✓		✓			✓					✓
ToxiProxy	Network	6					✓	✓		✓	✓	✓			✓
Istio	Kubernetes	2			✓		✓			✓	✓	✓			✓
KubeDoom	Kubernetes	1			✓	✓				✓			✓		✓
AWS FIS	Amazon Web Services (RDS, EC2, ECS, EKS)	7+	✓	✓	✓	✓	✓		✓					✓	✓

Our Weapon Selection: Litmus

- Cloud-Native Chaos Engineering Framework
- Open source, free and CNCF sandbox project
- Run chaos tests in a controlled way
- Provide ready to use chaos experiments
- Include a management portal
- Integrated with monitoring tools
- Rolling out changes using GitOps
- Manage users and teams
- Cross cloud support

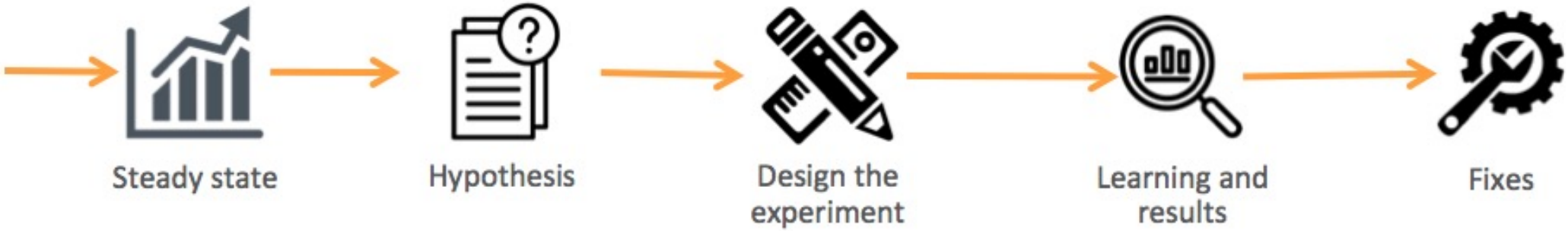
Let’s break things together!

Demo 02: Running Experiments (Litmus)



We will use Litmus to test our demo application: https://pacman.seladevops.com

Phases of Chaos Enginieering



It’s Chaos a Good Idea?

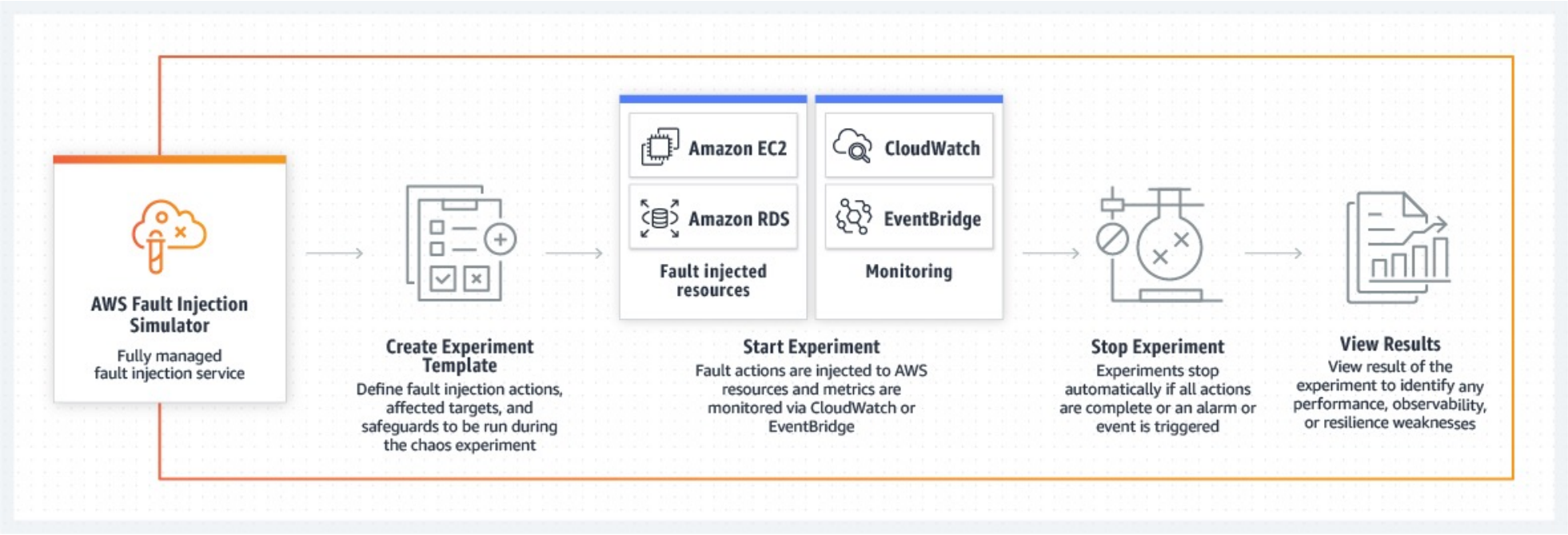
”Chaos doesn’t cause problems.
It reveals them”

Nora Jones

Senior Chaos Engineer, Netflix

Chaos Enginieering on AWS

AWS Fault Injection Simulator: Fully managed service for running fault injection experiments on AWS that makes it easier to improve an application's performance, observability, and resiliency.



Demo 03: Chaos Enginieering on AWS

Let’s use Fault Injection Simulator (FIS) to test our application:

1) **Steady State:** The application must always respond with http code 200
2) **Hypothesis:** If an outage occurs in the eu-central-1 region a failover mechanism will be activated and there will be minimal downtime for the end user.
3) **Experiment:** Simulate an outage by terminating all instances in region eu-east-1

Feel free to use the application during the experiment: https://pacman.seladevops.com

Demo Time

Demo 03: Chaos Enginieering on AWS

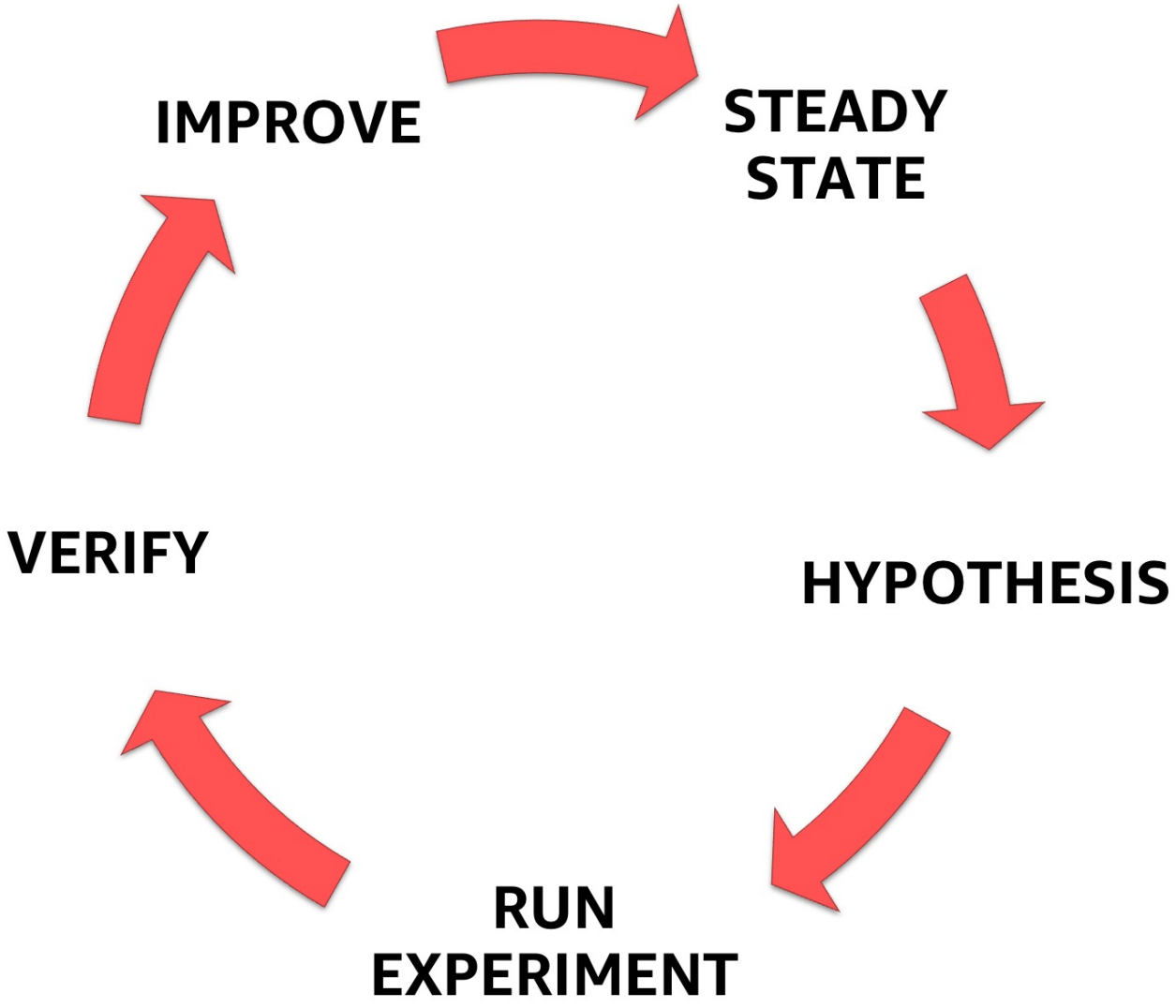
After our experiment we can conclude that:

1) **Results:** The failover mechanism was activated but there was some downtime for the application.
2) **Fixes:** It’s necessary to improve the mechanism in order to achieve zero downtime failovers.

Summary

- Chaos Engineering it's about running thoughtful, planned experiments in a controlled environment to intentionally break or disrupt an application based on a hypothesis of how you expect that application to fail.

- Chaos engineering is not about injecting failures randomly without an end goal in sight



Chaos Engineering

If you don't test your systems, they will test you when a failure occurs

Thank you for coming!

https://github.com/leonjalfon1/sdp-chaos