KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon
Europe 2022

Case Study: Bringing Chaos Engineering to the Cloud Native Developers

Uma Mukkara, Harness
Ramiro Berrelleza, Okteto

Case Study:

Bringing chaos engineering to to the cloud native developers

Uma Mukkara
Head of Chaos Engineering
Harness

Ramiro Berrelleza
CEO
Okteto

Service Reliability

Outages still happen

Service

System

Same system

Service are built with redundancy

Financial & Reputation Damage

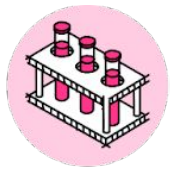
Service Downtime?

Metrics of importance:

MTTR

MTTF

Chaos Engineering Primer



Select Chaos Experiments
Ex: Simulate Region Goes Down, etc

2

1

3

Select systems to test

Chaos Engineering

Run Set of Chaos Experiments on Target System

5

4

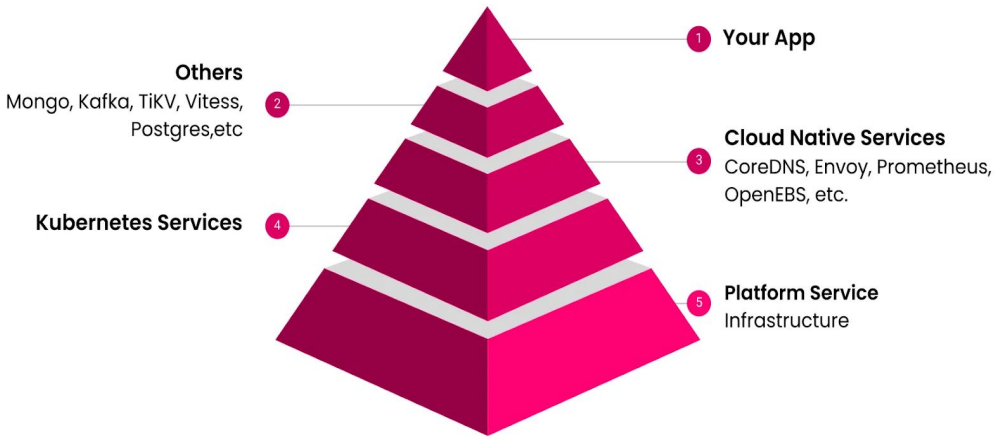
Use learnings to make targeted reliability improvements

Observe results of experiments on target system

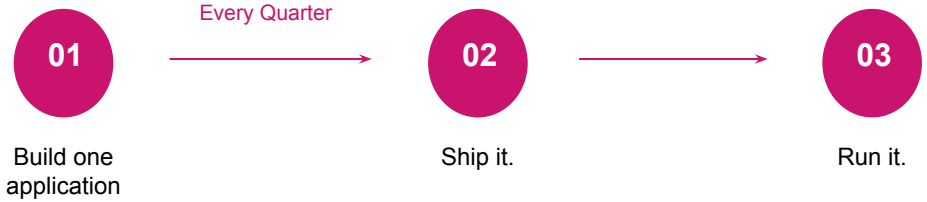
Why Chaos Engineering?

Proliferation of applications into micro services leads to a RELIABILITY challenge

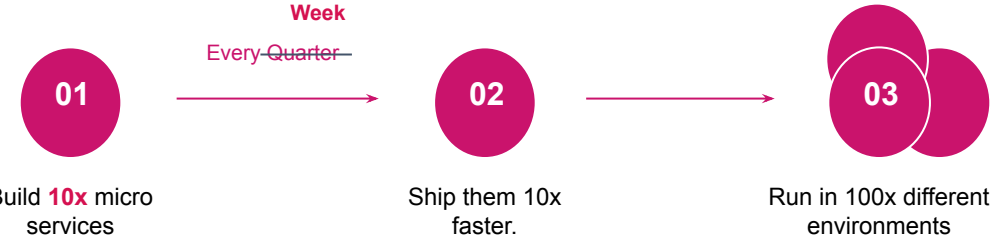
In cloud native, your code depends on hundreds of other microservices and runs on many platforms. The potential of being subjected to a dependent component failure is huge.



Legacy DevOps

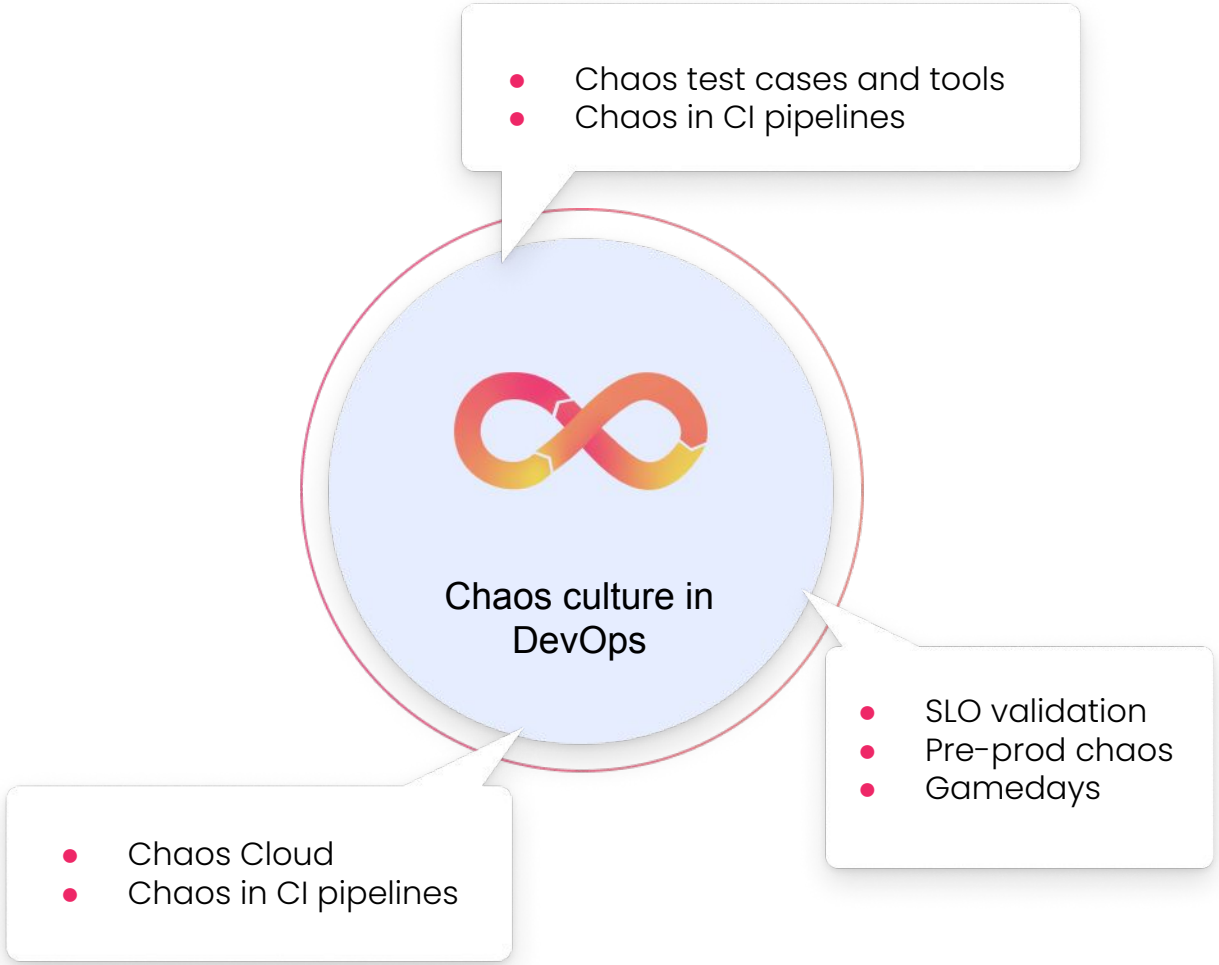


Cloud native DevOps



Too many fault scenarios. Significant increase in service down potential because of a failure of a dependent service

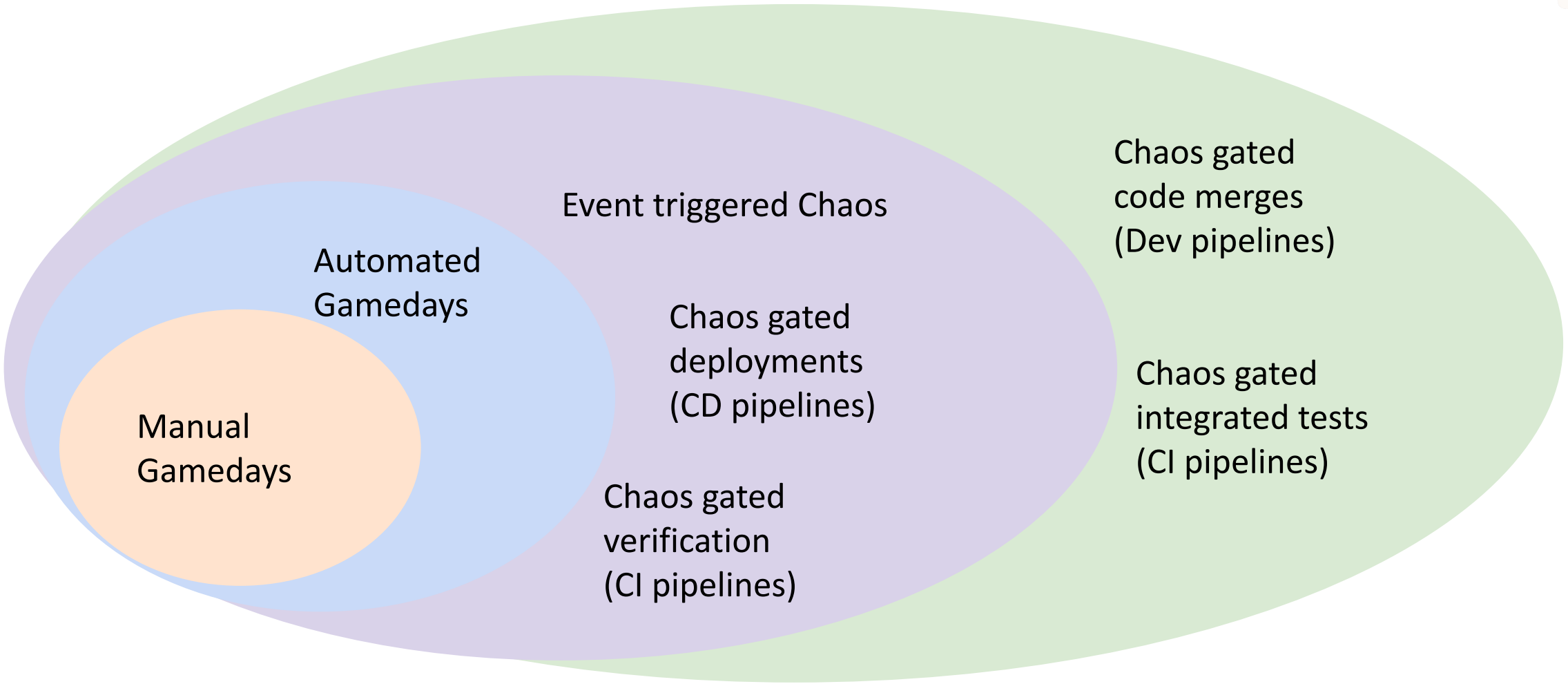
Chaos Engineering is a DevOps culture



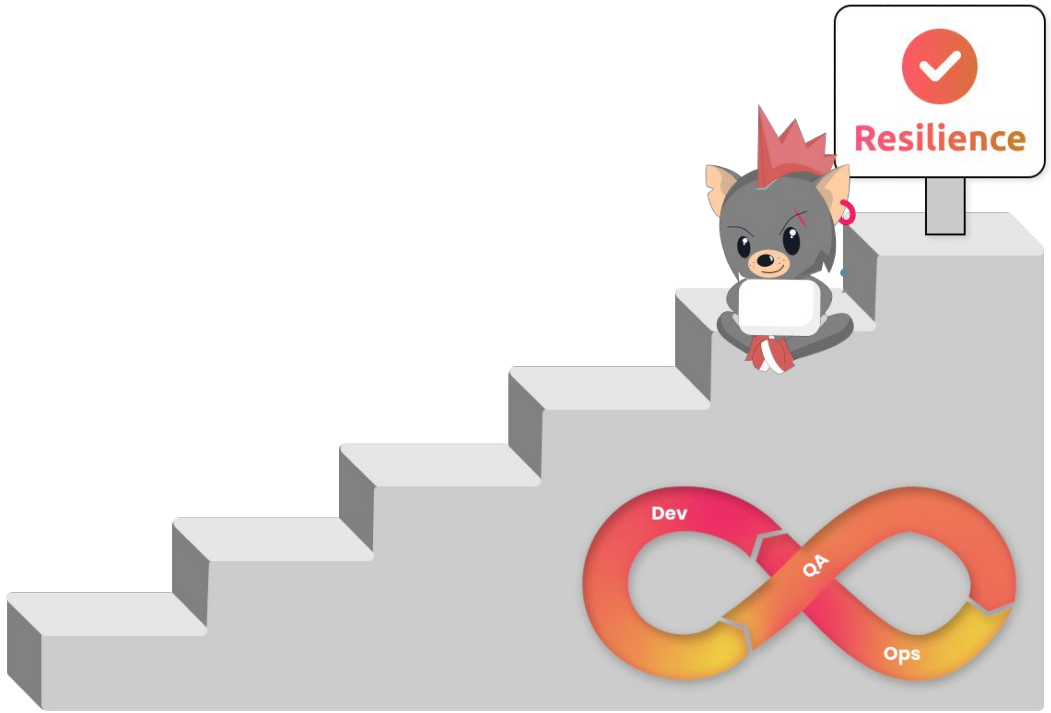
adopts chaos culture in cloud native DevOps

- Bring chaos tools for developers. Make chaos execution a natural extension of cloud native development.
- Integrate chaos capabilities into the test/QA flow.
- Make chaos easy to use for Ops. Integrate with observability tools and CI/CD platforms.

Chaos Engineering Maturity levels - Personas and areas



Developers and Chaos Engineering - Why?



Dev – ROI

- Developer ROI - Test code against potential failures in hundreds of microservices before merging.
- QA ROI - Test product against potential failures in hundreds of microservices in the user acceptance tests (before CD)

Improved CI/CD efficiency

Developers and Chaos Engineering - How?

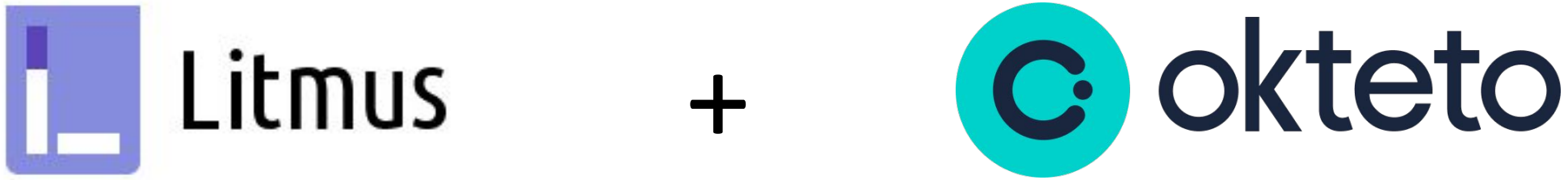
Let’s see through a demo

Litmus

LitmusChaos is an open source Chaos Engineering platform that enables teams to identify weaknesses & potential outages in infrastructures by inducing chaos tests in a controlled

Okteto is an open source tool that enables team to quickly spin up Kubernetes-based development environments in order to speed up their development process

Demo



Let's make Chaos Engineering a Dev + DevOps culture

By giving developers access to remote development environments, we make it easier to run Chaos experiments all the time.

We eliminate the perception that Chaos testing is expensive. All it takes is to run a script!

As developers run more and more Chaos experiments, they get better at it.

Let's make Chaos Engineering a Dev + DevOps culture

Bring chaos tools for developers. Make chaos execution a natural extension of cloud native development.

Make chaos an obvious part of the Development process. Don't put chaos behind the CI/QA gate

Quality will go up, and the need for rework will go down. Everybody wins.

Q&A

github.com/litmuschaos.io/litmus

github.com/okteto/okteto