



KUBERNETES COMMUNITY DAYS CHENNAI 2023

22 July, Chennai Trade Centre, Tamil Nadu

Reliability in Cloud Native Services

Keynote - KCD Chennai 2023

22nd July 2023

Uma Mukkara, Head of Chaos Engineering at Harness

About Me



Uma Mukkara

Head of Chaos Engineering at Harness

Co-founder and Maintainer of LitmusChaos



Agenda

Reliability in DevOps

What are the challenges in DevOps and how Reliability is getting affected?

Myths and Facts of Chaos

What are the commonly misunderstood items about Chaos Engineering

Chaos Adoption Best Practices

Where to start and proceed

Continuous Resilience

Traditional vs Modern approach to building Resilience

Chaos Maturity Model

The 4 levels of Chaos Maturity

Reliability in DevOps

Today's Challenges in DevOps

SPEED

QUALITY

DEVELOPER
PRODUCTIVITY



Today's Challenges in DevOps



Teams are tasked to ship the changes FAST. A lot of investments in modern tools and architectures are happening.

Move to CloudNative

Today's Challenges in DevOps



Invest heavily into
Automation

The number of components and
scenarios you are dealing with is
ever increasing. How to retain the
quality of the code being shipped
on a daily basis?

Today's Challenges in DevOps

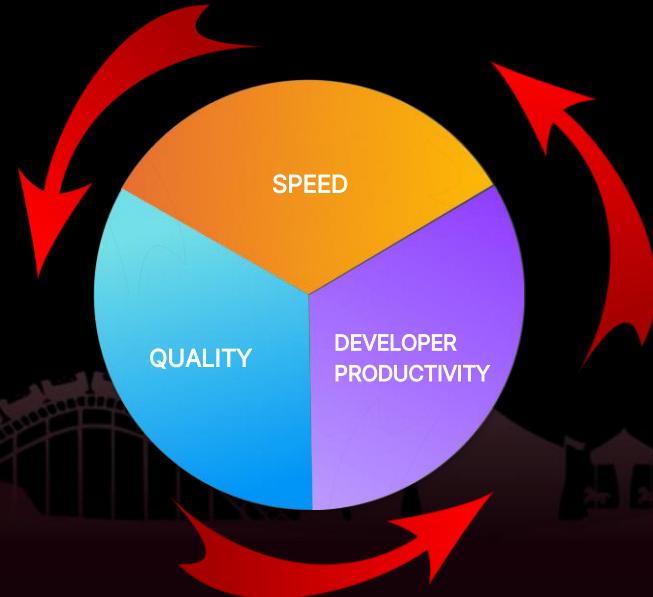


Give modern tools to
Developers

Developers are in firefighting
mode most of the time. Speed vs
Quality vs Tech Debt

Side effect of this innovation

NOT enough focus on Reliability



Is Reliability a goal in DevOps?



It is not a direct goal usually, but Reliability of the end product or service is being affected while solving the other challenges.



Are you sure you are not compromising the reliability?



Have you verified that the known resilience status is intact? No new bugs being leaked into the product?



How much of developer time is being spent on issues related to reliability?

Retaining reliability
can certainly be
a goal in DevOps



How to retain reliability?



The obvious questions are

1

Reliability of what?

2

Reliability against what?

3

How do you measure it?



The obvious questions are

1

Reliability of what?



Reliability of business services

2

Reliability against what?



Reliability against various types of faults that may occur

3

How do you measure it?



Steady state checks

The obvious questions are

1

Reliability of what?



Reliability of business services

2

Reliability against what?



Reliability against various types of faults that may occur

3

How do you measure it?



Steady state checks

Reliability of business services

Reliability against various types of faults that may occur

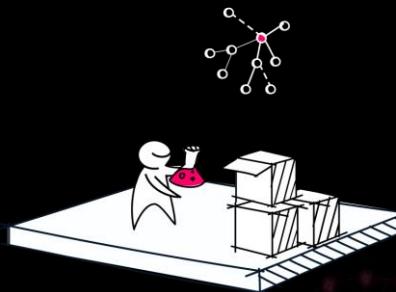
Steady state checks

Measure the steady states of business critical services against various types of faults

Measure the steady
states of business
critical services
against various types
of faults



CHAOS
ENGINEERING



Tackle Reliability with Chaos Engineering

You can **increase** reliability

You can **retain** reliability



Traditional vs Modern systems

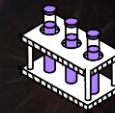
(monolith)

A natural choice Now

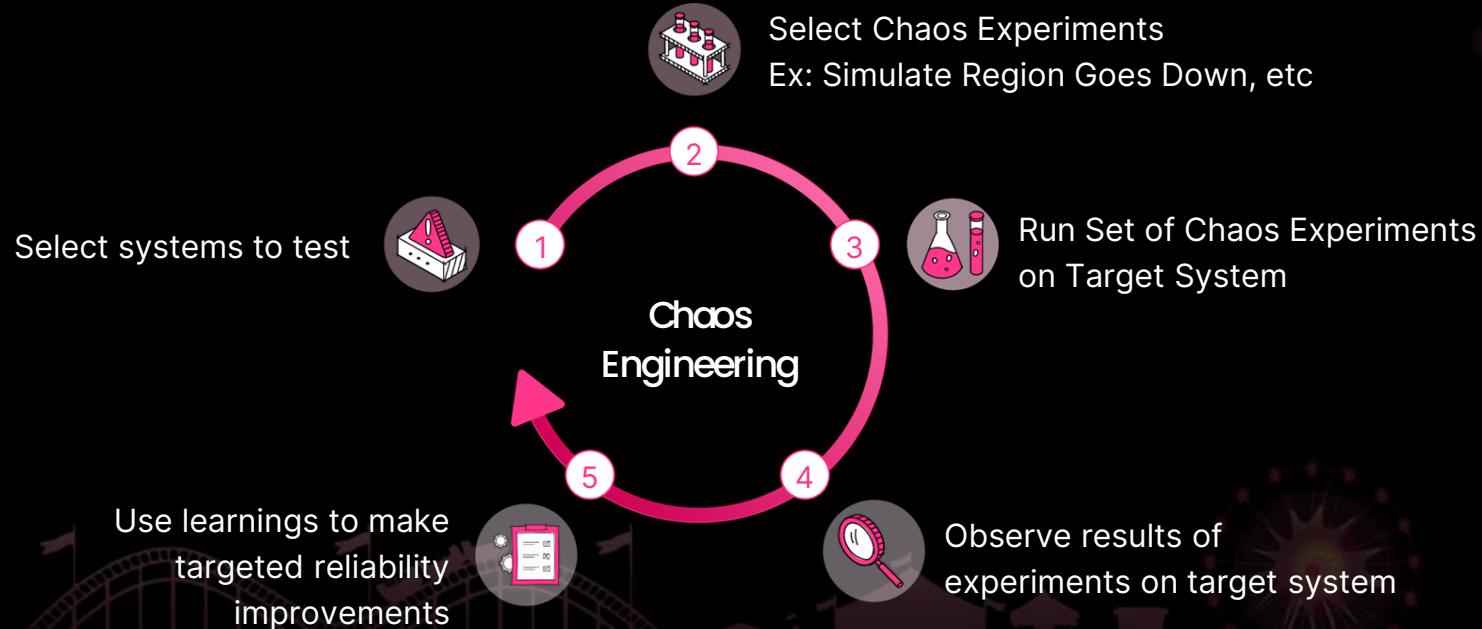
(K8s platform, containers, load, scale, cloud platforms)



Chaos Engineering is the **efficient** way to approach reliability
(Saves cost, effort and time)



Chaos Engineering 101



Who is using Chaos Engineering today?

Financial services



For regulatory reasons and for real reliability management under high loads or new technology

Highly scaled environments



Unexpected reliability issues are common under high loads. Retail, Fintech, Telecom etc

DR scenarios



A proof of DR testing that can be audited and serve the regulatory requirements

Kubernetes environments



Kubernetes architecture allows a lot of faults to happen naturally. Functional validation with random fault injection is a natural need in QA

Implementation challenges

- ◆ Don't know what to test?
- ◆ How to measure reliability score?
- ◆ What is the maturity model?
- ◆ Don't know who will create and who will run?
- ◆ How to measure how much we covered?
- ◆ What is the ROI I can promise?



Myths and Facts about Chaos Engineering

Myth #1

Chaos Engineering and Failure testing are same



Fact

Chas Engineering is much wider
than Failure Testing



Fact

Chas Engineering is much wider than Failure Testing



- Failure testing is related to functional testing
- Chaos Testing is about checking if the end service continues to be reliable or not. Thinking about the bigger picture.
- A larger process is involved to discuss, debate and design the steady state.

Myth #2

Chaos Engineering is for SREs only



Fact

CE is for QA and Developers too



Fact

CE is for QA and Developers too



- QA in cloud native is becoming more complex with too many micro services to handle. The underlying stack is also containerized
- Chaos Engineering is almost becoming a need for QA teams
- Left shift is happening faster than imagined

Myth #3

Chaos Engineering is about injecting faults



Fact

Injecting Faults and Steady State deviation observation,
being smart about finding weaknesses. Engineer a
process to root out weaknesses.



Fact

Injecting Faults and Steady State deviation observation, being smart about finding weaknesses. Engineer a process to root out weaknesses.



- Injecting faults is only one part of the engineering.
- Steady state observation is often the first benefit of chaos engineering
- You may have APM systems, but are you really watching the steady state deviation?

Myth #4

Chaos Engineering is not expensive



Fact

Not inexpensive. Needs budgeting. Continuous budgeting. Real results take time and resources. It is a cultural change, comes with some investment.



Fact

Not inexpensive. Needs budgeting.
Continuous budgeting. Real results take time
and resources. It is a cultural change, comes
with some investment.



- One person practicing chaos engineering will not yield cultural shift in DevOps
- Unbudgeted Chaos Engineering efforts are often the reason for the failure in ROI
- Automation and increasing chaos coverage require resources

Myth #5

Chaos Engineering is quick



Fact

It is a continuous process. Takes time for real results. Some results can be achieved in short term. But takes time to achieve maturity and proven results at scale.



Fact

It is a continuous process. Takes time for real results. Some results can be achieved in short term. But takes time to achieve maturity and proven results at scale.



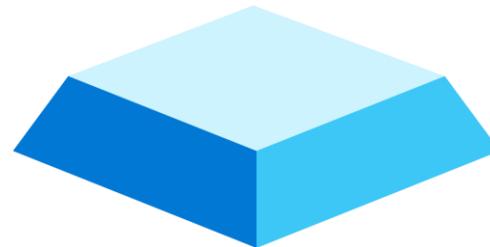
- Chaos engineering in an organization will go through the maturity cycles. Time and money - both are needed
- Anywhere from 1 year to a few years for achieving maturity
- Large organizations take more time as often the budgets play the spoilsport

Chaos Engineering Adoption Patterns



Target Pattern

Infrastructure Chaos





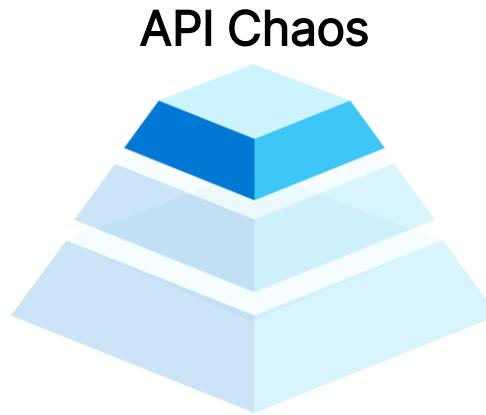
Target
Pattern

Resource Chaos





Target Pattern





Target Pattern

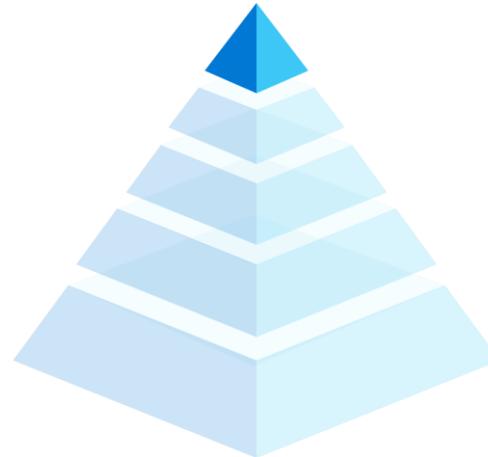
Application Chaos





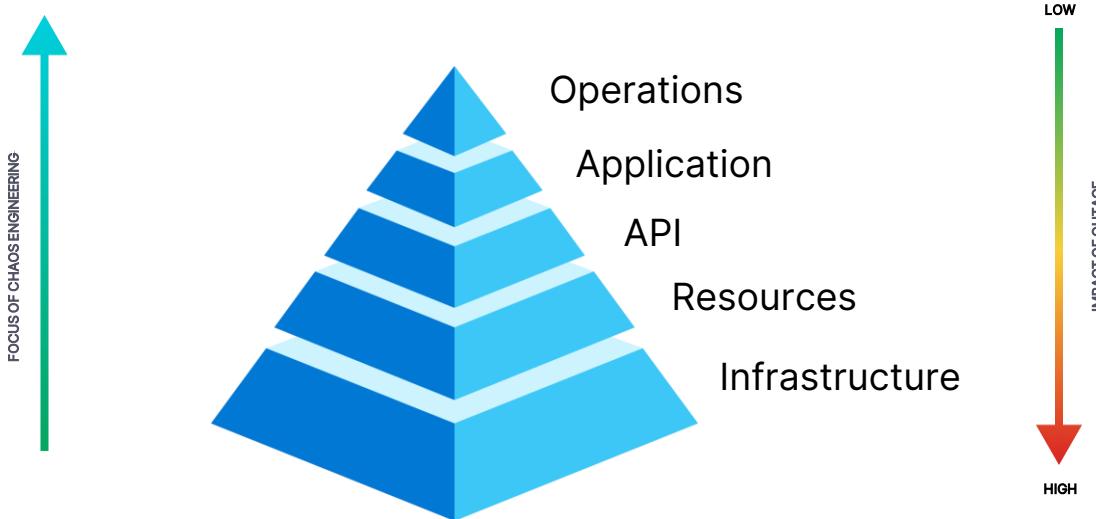
Target
Pattern

Operations Chaos





Impact & Focus

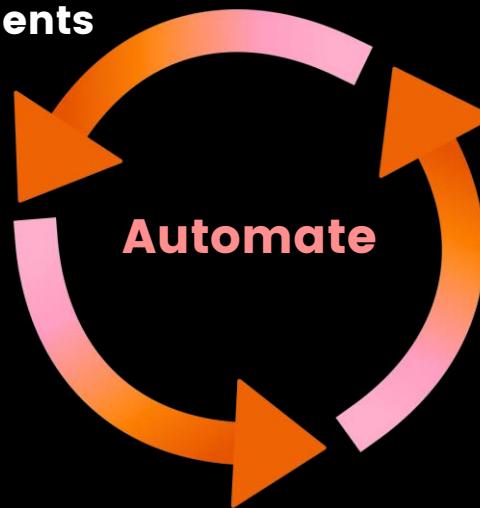


Continuous Resilience

**Automation
is the key
for
productivity
& outcomes**

**Creation of the
new experiments**

**Recovery
procedures**



Experiment runs

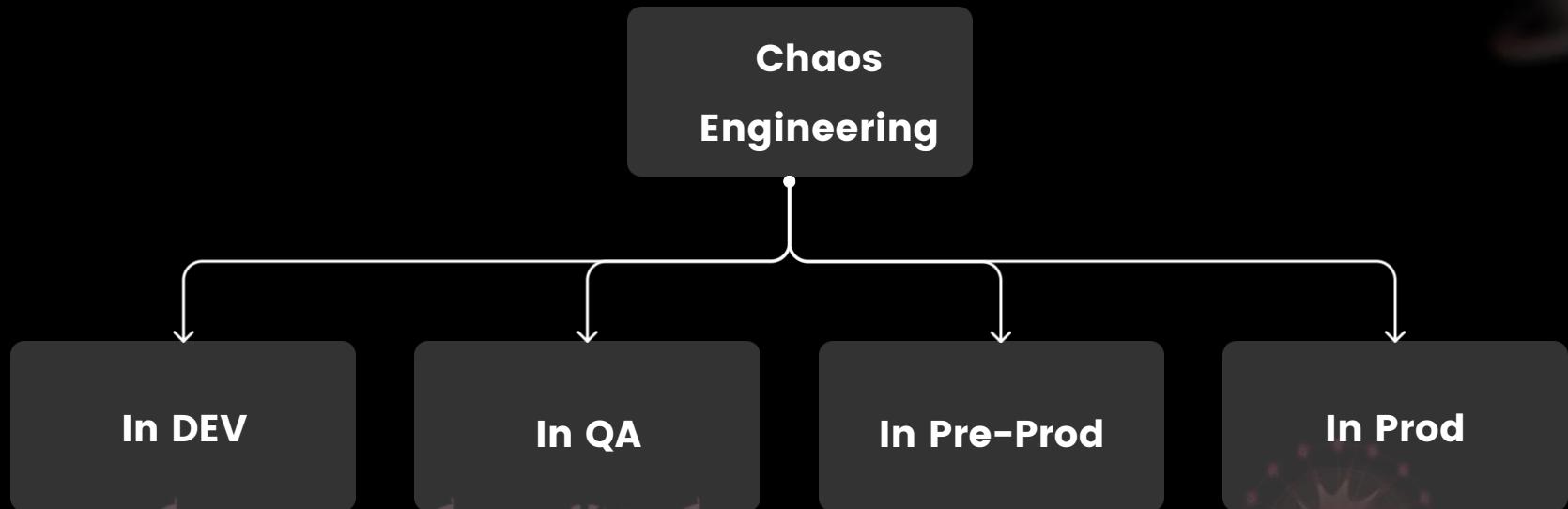
MODERN CHAOS ENGINEERING

Continuous

Verifying resilience through automated
Resilience Chaos Testing Continuously



Continuous Resilience



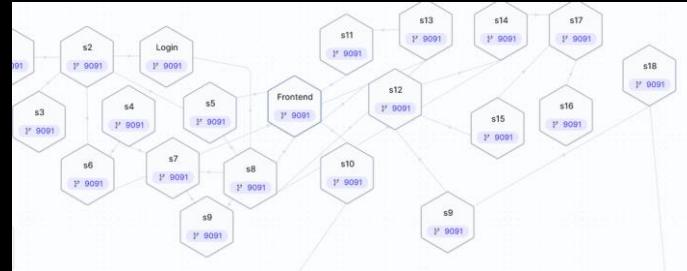
Continuous Resilience Metrics

The Regular Metrics for Reliability:

MTTF

MTTR

MTTI



New Metrics are:



Resilience Score

The average success % of steady state of a given experiments or a component or service



Resilience Coverage

The number of chaos tests executed

Total number of chaos tests possible

X 100

Gameday vs Pipeline Approach

Chaos via Game Days

Chaos Experiments are executed on demand and with a lot of preparation

Primarily targeted towards SRE as a persona

Adoption barrier is very high

Chaos via Pipelines

Chaos Experiments are executed continuously and without much preparation

All personas are executing the chaos experiments

Adoption barrier is much less

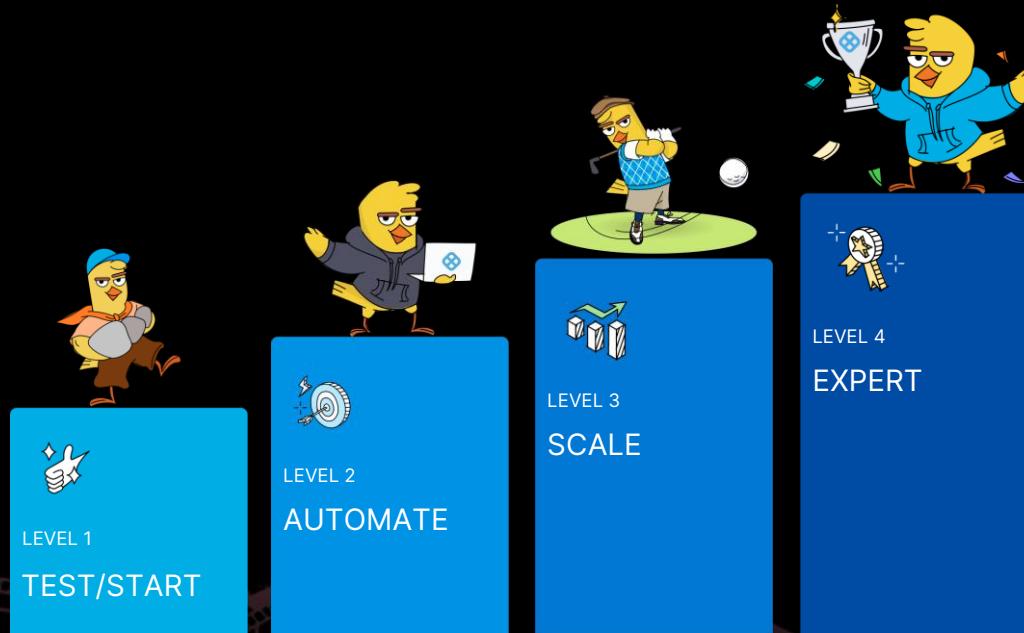
VS

Chaos Maturity

Model



Chaos Maturity Model



Summary



Resilience building is a continuous activity

- Build it in pipelines and automated testing in production



**Start Chaos Engineering with a few developers in QA
with one network map or a business service and expand
it to the entire organization via Chaos Hubs**

OPEN SOURCE



ENTERPRISE



Thank you

