steadybit

Reliability in the Face of Uncertainty

Benjamin Wilms
CEO & Co-Founder

@MrBWilms

linkedin.com/in/benjamin-wilms

16 February 2023 — Conf42 Chaos Engineering

What is the mission of software development?

Continuously improve & deliver a software solution that reliably delivers value to its users.

What is reliable?

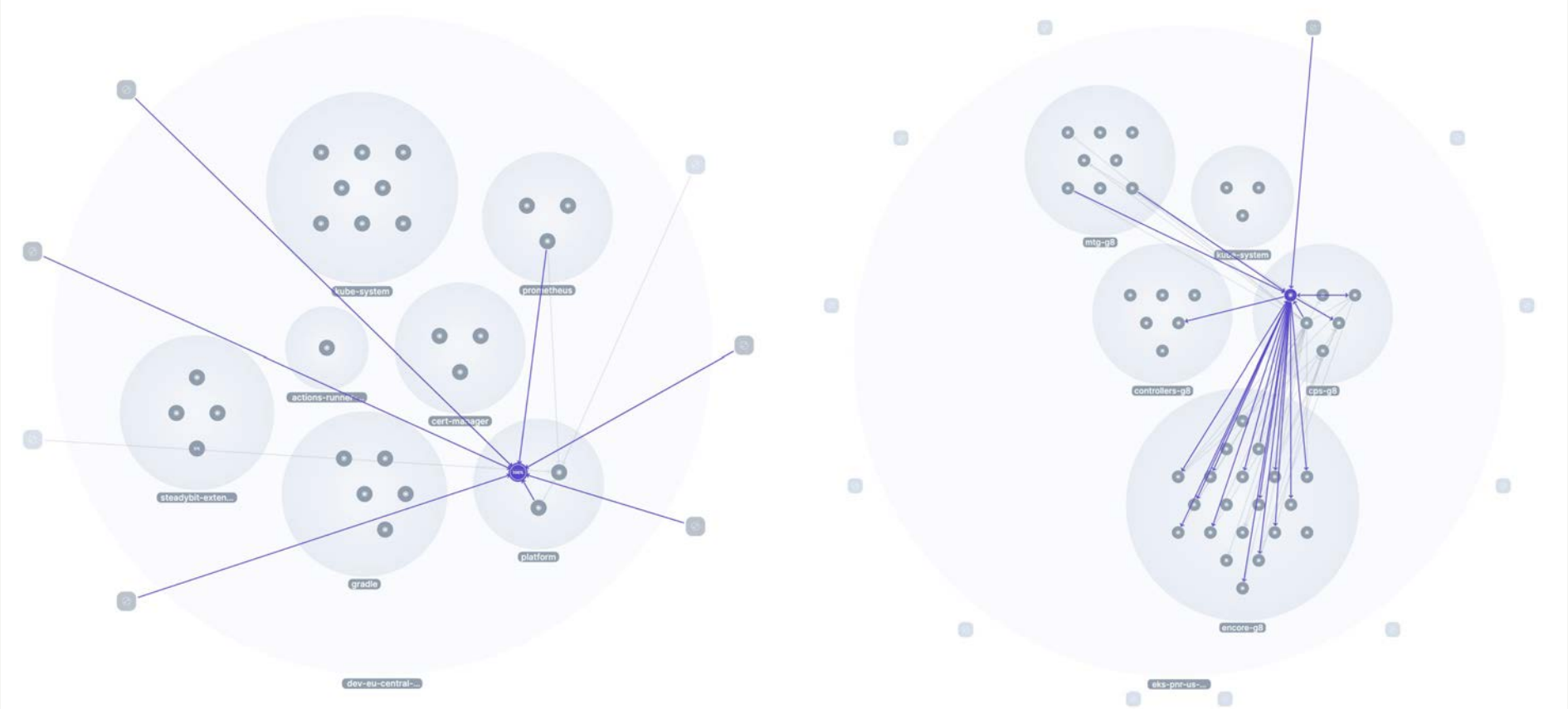
Reliable = consistently good in quality and performance

Reliable = able to be trusted

We trust a system when it’s consistently good in quality and performance

Today’s Systems

Today’s Systems



It’s not surprising that your system sometimes fails.

Dr. Richard Cook

It’s not surprising that your system sometimes fails.

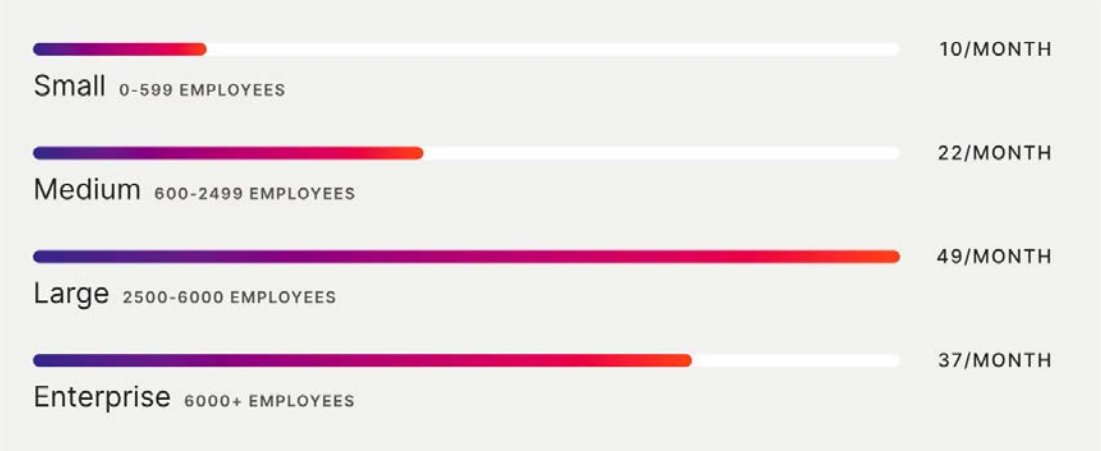
What is surprising that it ever works at all.

Dr. Richard Cook

We haven’t solved this yet

Dr. Richard Cook

Incidents by company size



This report is based on 53,034 incidents resolved on the FireHydrant platform between 2019 and 2022.

Average time to resolve incidents

24
HRS

05
MINS

This report is based on 53,034 incidents resolved on the FireHydrant platform between 2019 and 2022.

Everything fails all the time

Werner Vogels, VP & CTO Amazon

What's normal?

Failures are normal

Under those chaotic conditions...

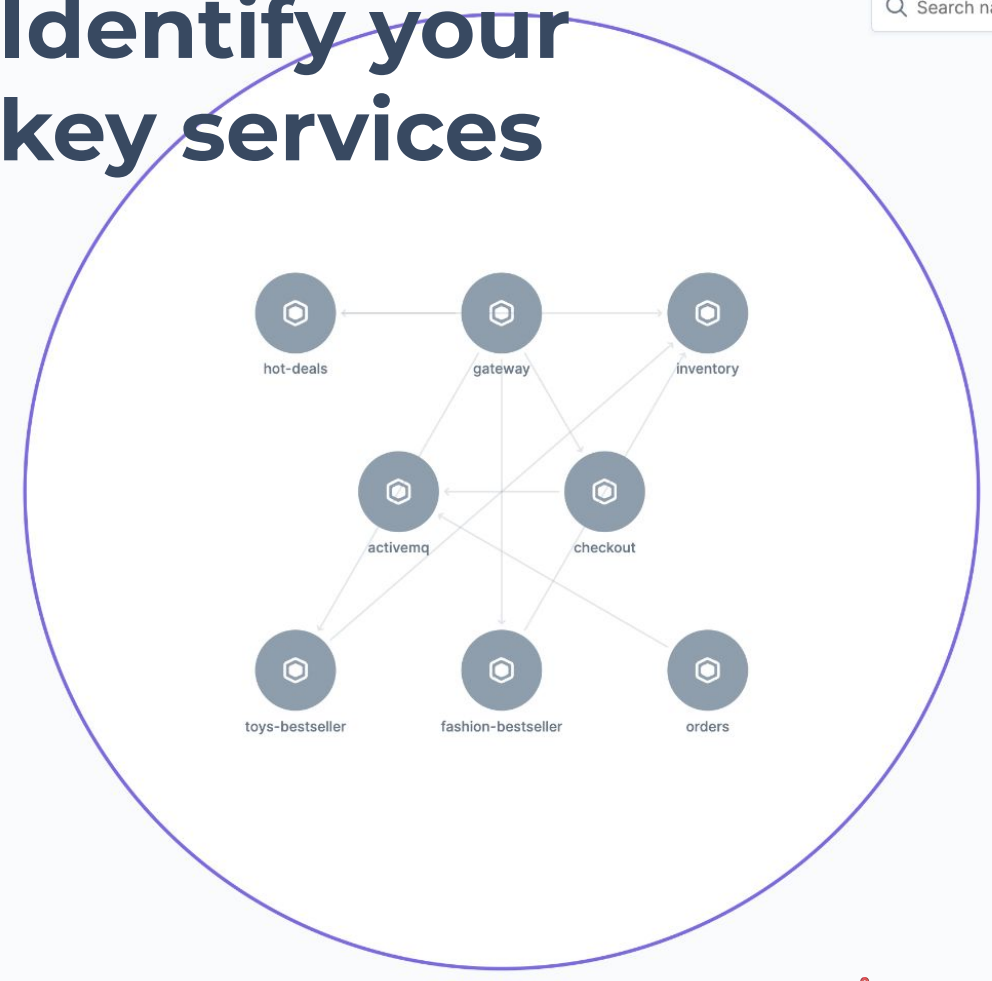
Chaos Engineering is necessary

Failures are the foundation of experiments

What can we do?

Proactively improve the reliability of your system

Identify your key services



Search namespace, deployment,...

demo-prod /

steadybit-demo
Namespace

Deployments 8

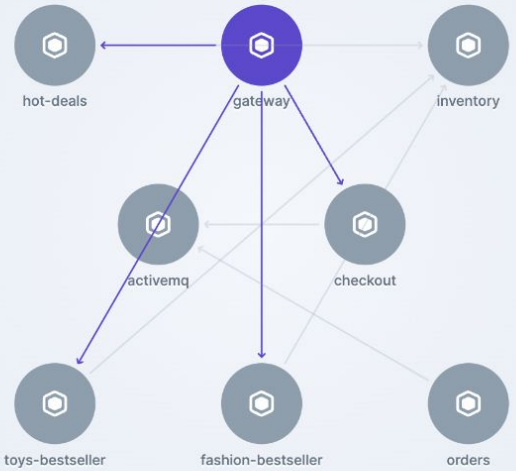
Name	Ready
activemq	2 / 2
checkout	2 / 2
fashion-bestseller	1 / 1
gateway	2 / 2
hot-deals	2 / 2
inventory	2 / 2
orders	2 / 2
toys-bestseller	2 / 2

steadybit-demo

steadybit

160 %

Identify your key services



Search namespace, deployment,...

... / steadybit-demo

gateway
steadybit-demo

Kubernetes Information

Labels
run=gateway

Avg. time to readiness
120s

Pods 2

gateway-587b9fc7fc-tsq68	Running (ready)
gateway-587b9fc7fc-kblmr	Running (ready)

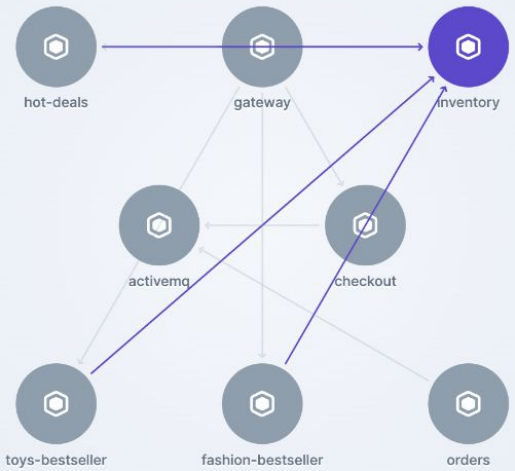
Bind Policies

steadybit-demo

steadybit

160 %

Identify your key services



Search namespace, deployment,...

... / steadybit-demo

inventory
steadybit-demo

Kubernetes Information

Labels
run=inventory

Avg. time to readiness
101s

Pods 2

inventory-56cf8fddcd-6r2vc	Running (ready)
inventory-56cf8fddcd-6xhhp	Running (ready)

Bind Policies

steadybit-demo

steadybit

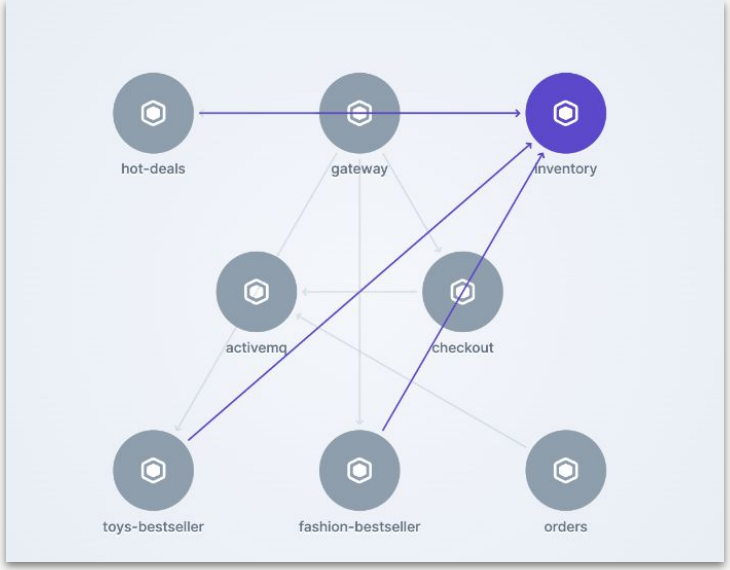
160 %

How can we do this proactively?

Test under real conditions early as possible

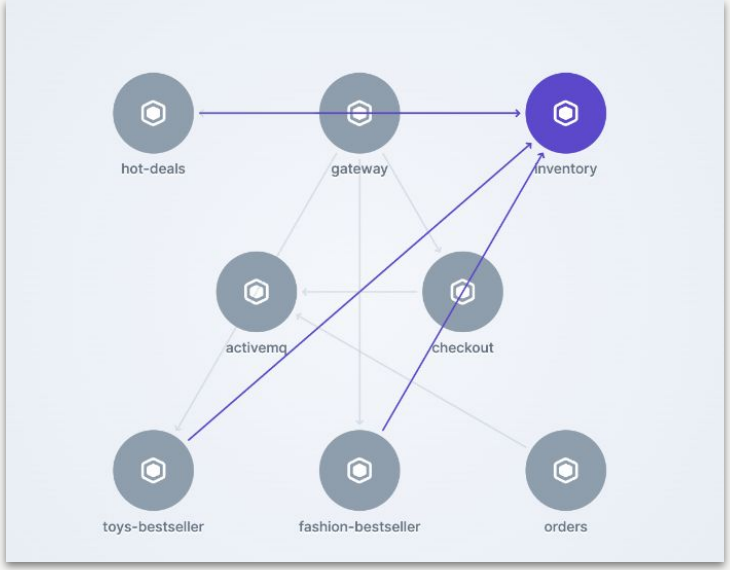
Real Conditions

Normal *Inventory* response time is 25ms

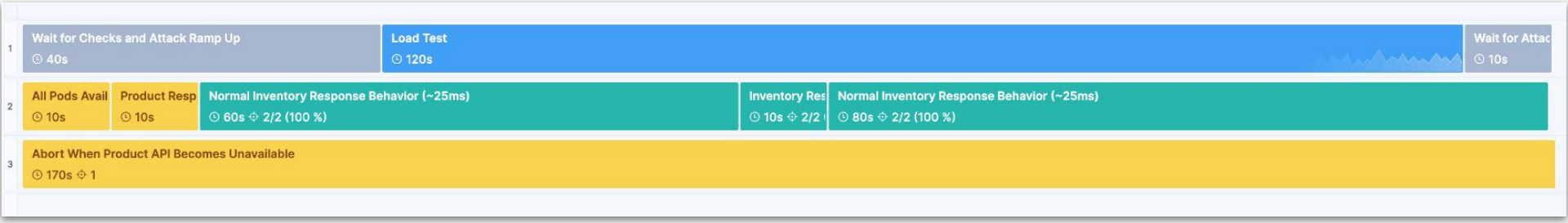


Real Conditions

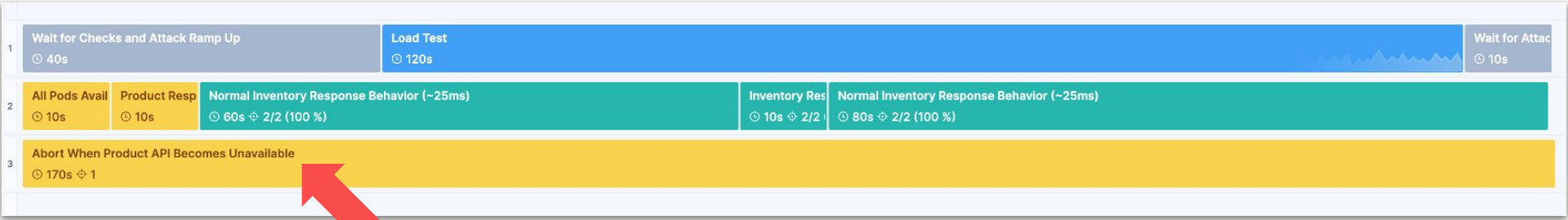
Inventory response time spikes up to 500ms



Experiment Design

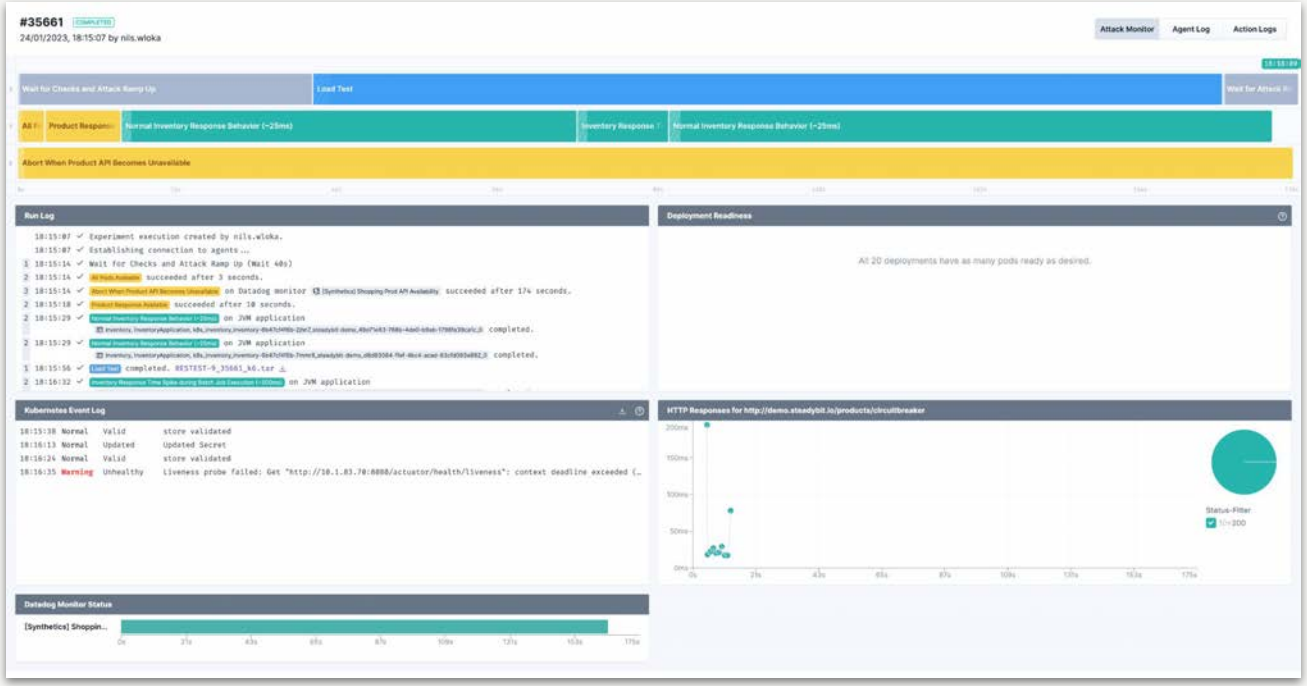


Experiment Design



Conditions under which the experiment fails

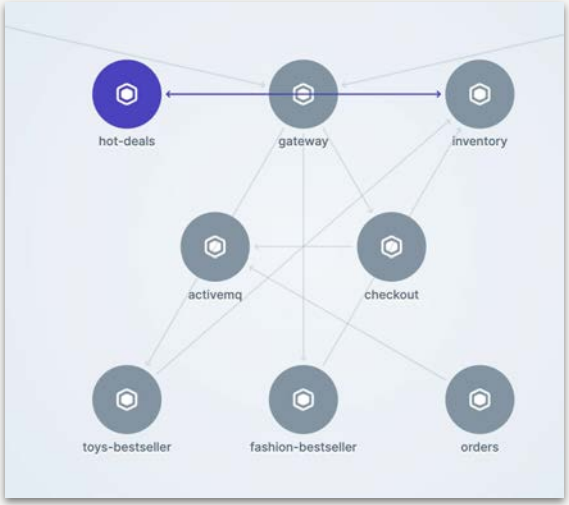
Experiment Run



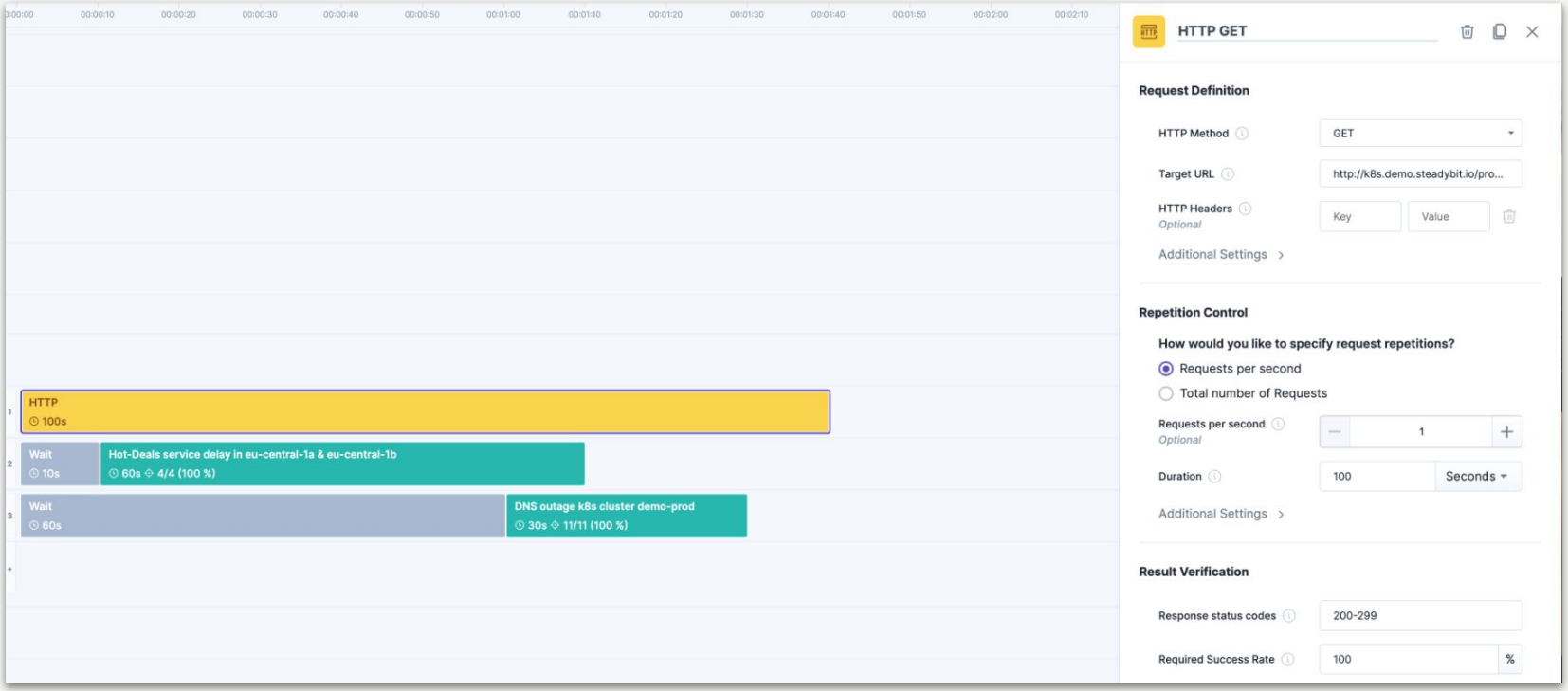
steadybit

Real Conditions

Hot-Deals response delay 2 zones
followed by a DNS outage in our Kubernetes cluster

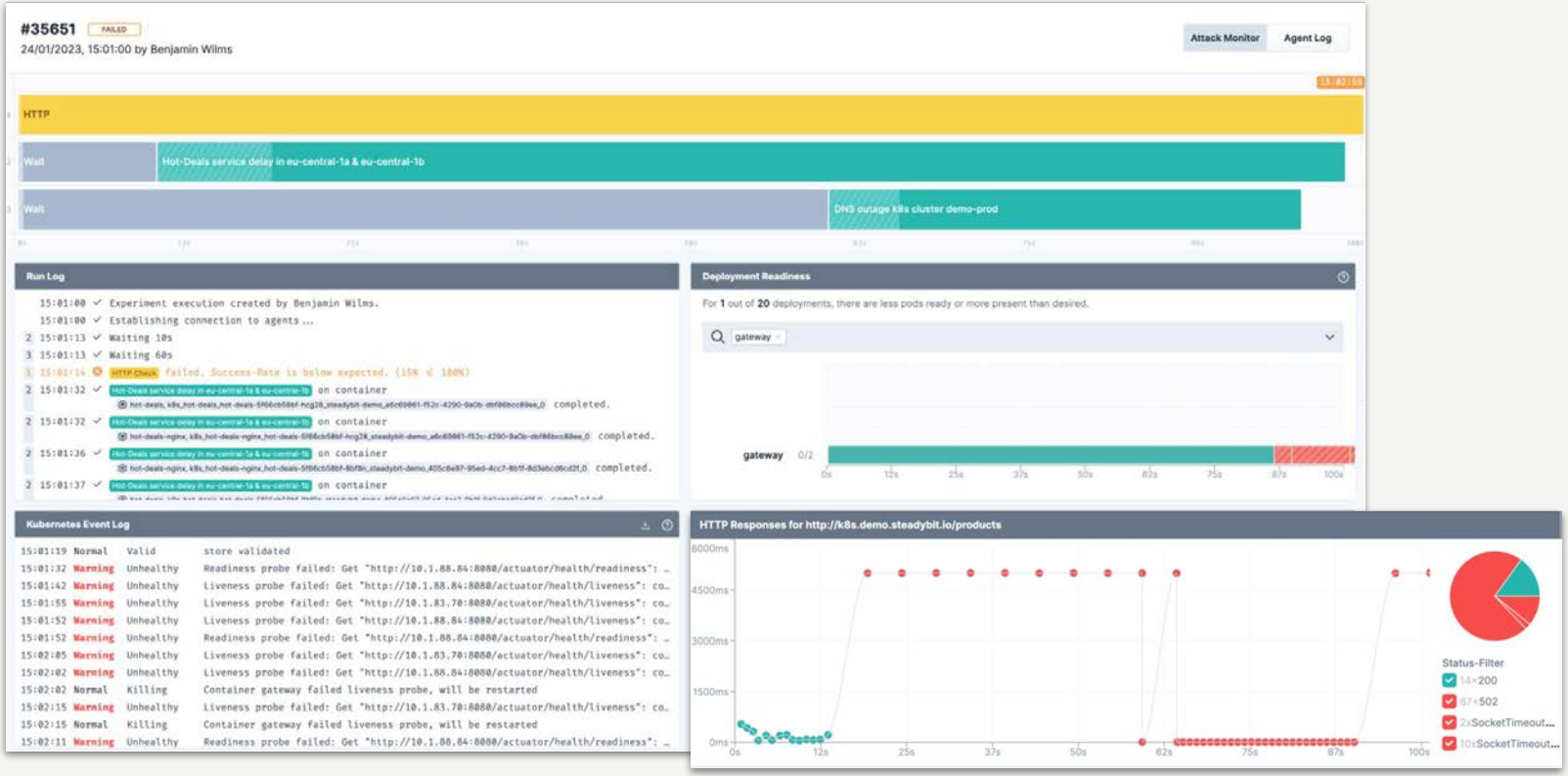


Experiment Design



steadybit

Experiment Run

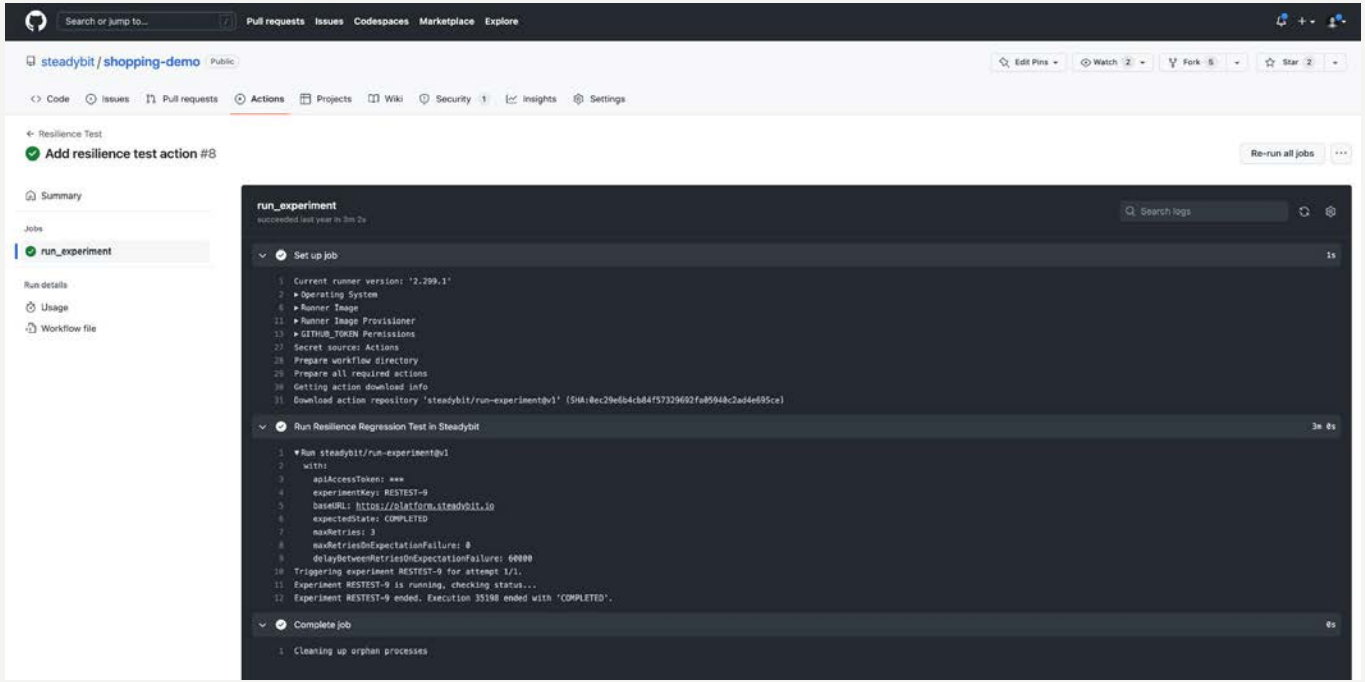


steadybit

When should we run these experiments?

Make it an integral part of the software development process

GitHub Action

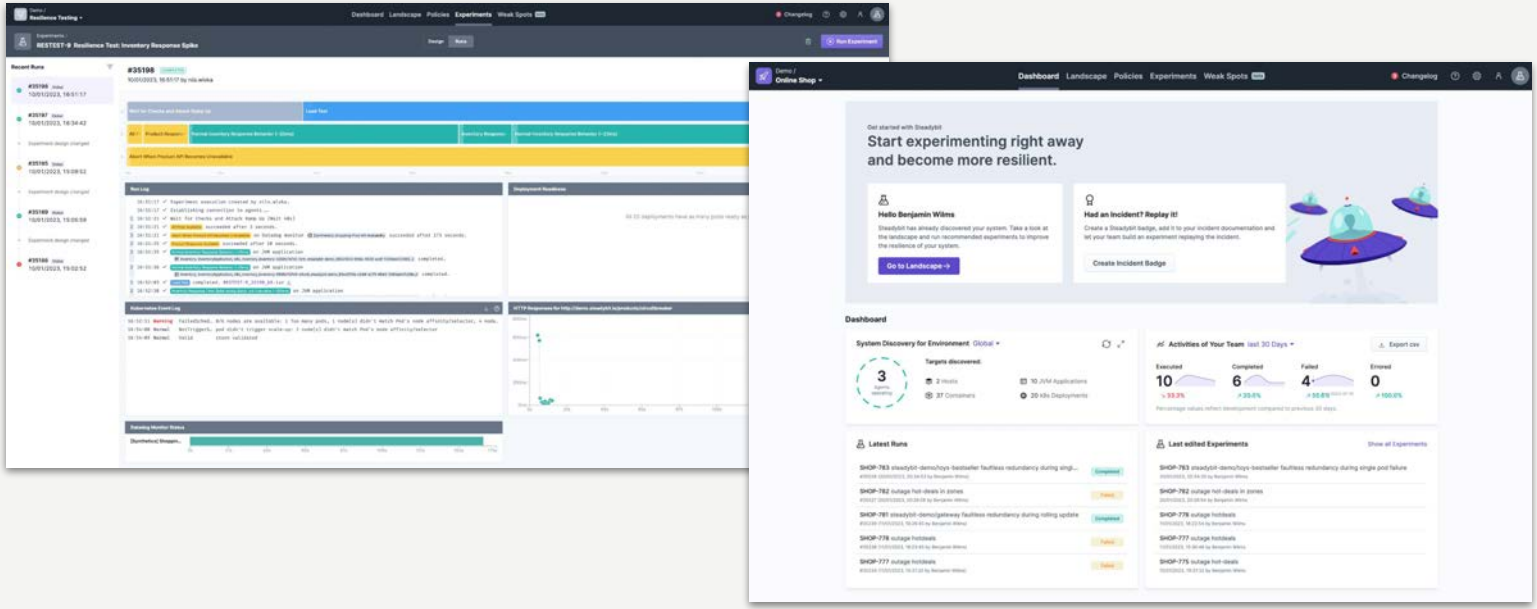


steadybit

Recap

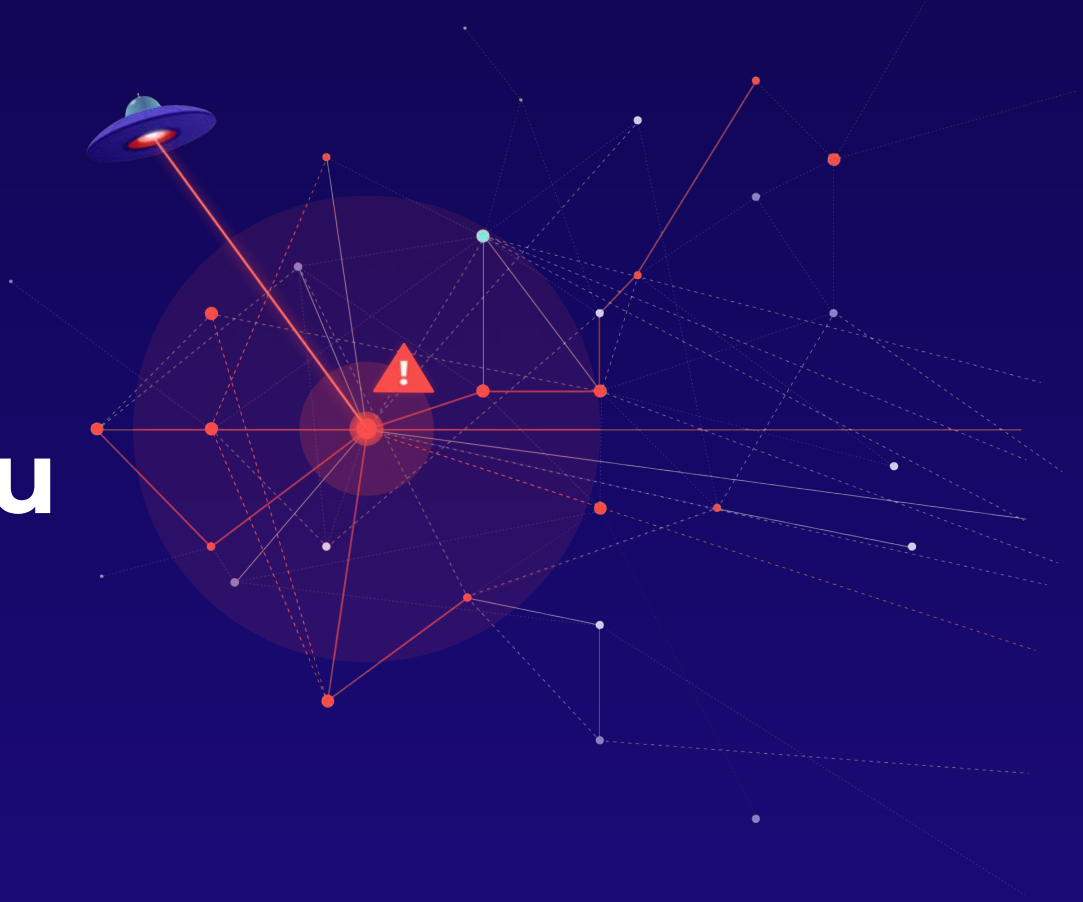
Embrace failures and turn them into experiments

Chaos Engineering with Steadybit



steadybit

Thank you



steadybit.com