



# Reliability in the Face of Uncertainty



**Benjamin Wilms**  
CEO & Co-Founder

 @MrBWilms

 [linkedin.com/in/benjamin-wilms](https://www.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

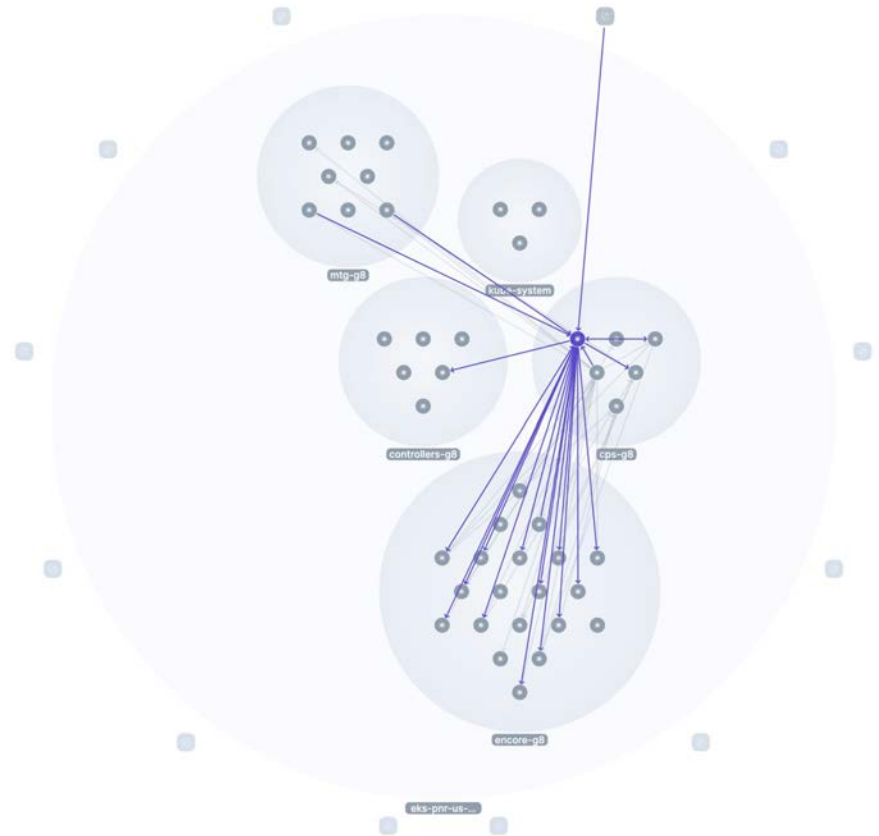
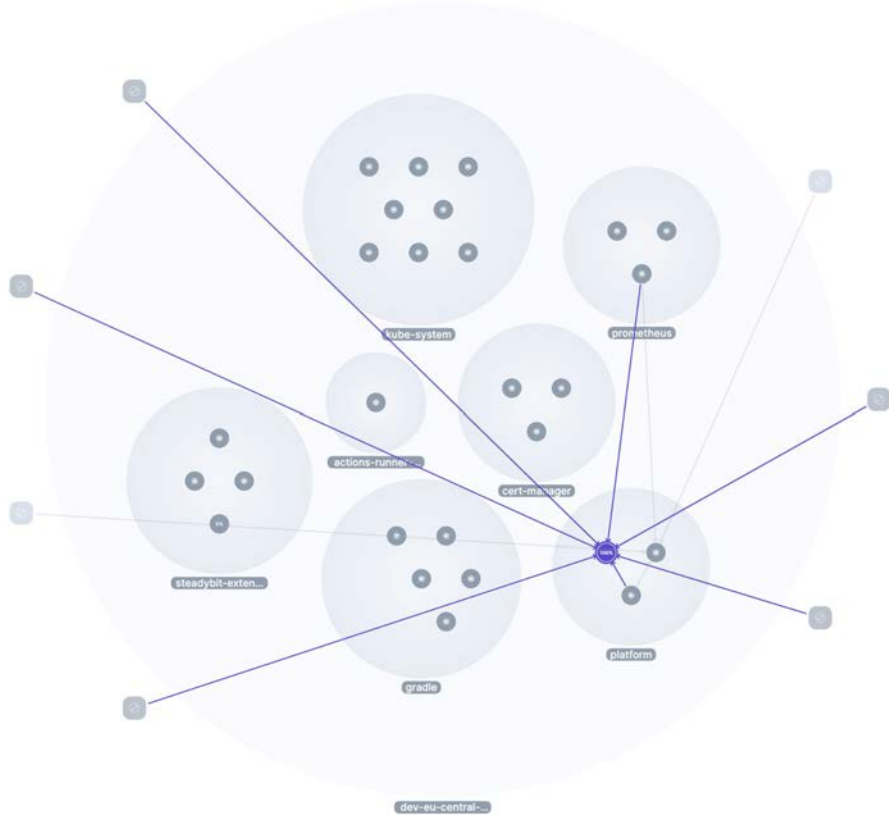
**Reliable** = able to be trusted

**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.***

It's not surprising that your system  
sometimes fails.

***What is surprising that it ever works  
at all.***

**We haven't solved this yet**

# 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...**

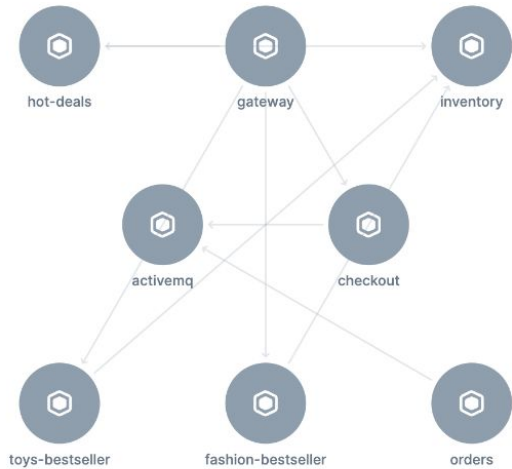
# Chaos Engineering is necessary

**Failures** are the foundation of  
experiments

# What can we do?

**Proactively** improve the reliability of  
your system

# Identify your key services



steadybit-demo






Search namespace, deployment,...

demo-prod /

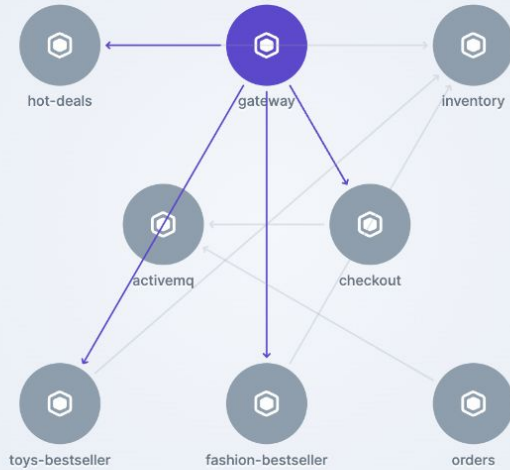


steadybit-demo  
Namespace

## 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

# Identify your key services



Search namespace, deployment,...

... / steadybit-demo

gateway  
steadybit-demo

## Kubernetes Information

### Labels

run=gateway

### Avg. time to readiness

120s

### Pods 2

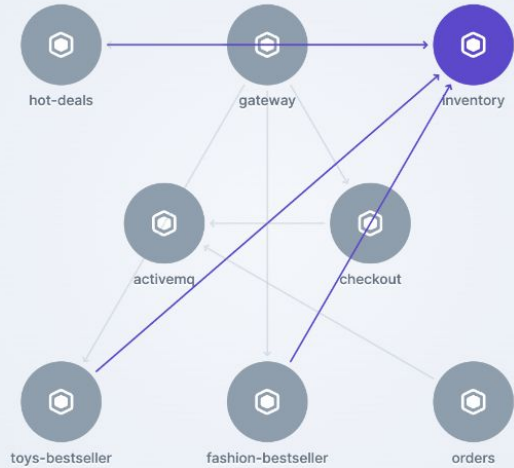
gateway-587b9fc7fc-tsqq8	Running (ready)
gateway-587b9fc7fc-kblmr	Running (ready)

Bind Policies

steadybit-demo



# 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

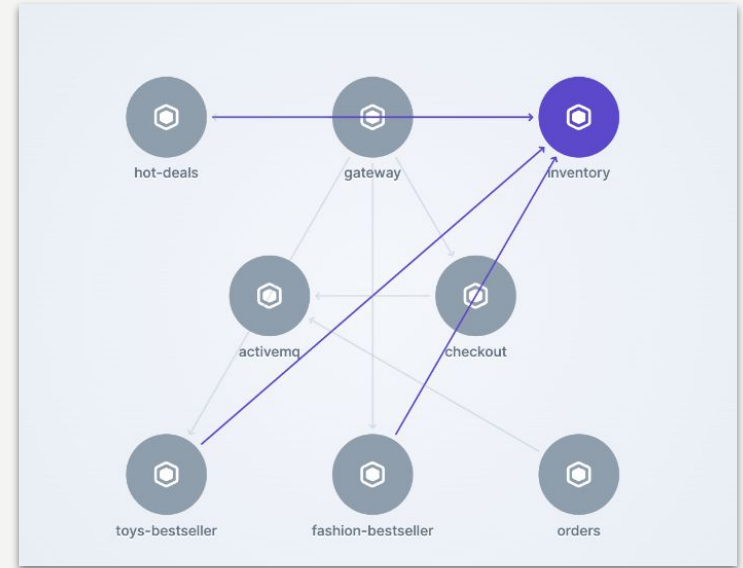
160 %

**How can we do this proactively?**

**Test under real conditions early as possible**

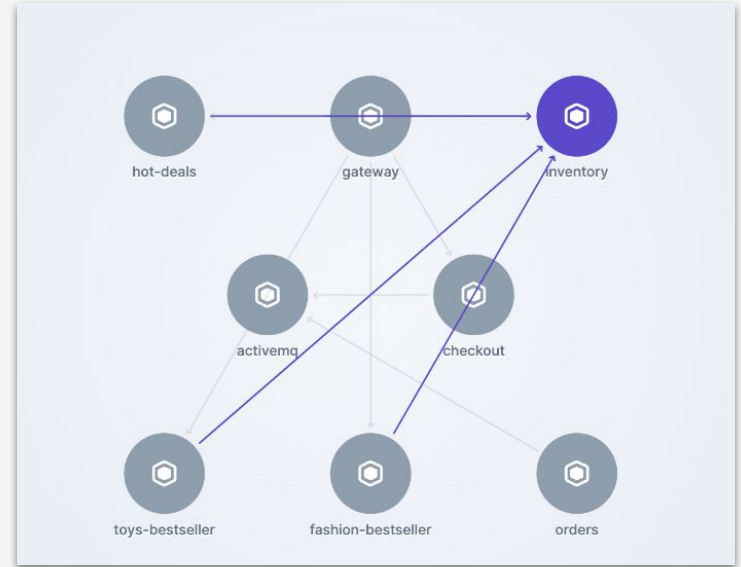
# Real Conditions

Normal *Inventory* response time is 25ms

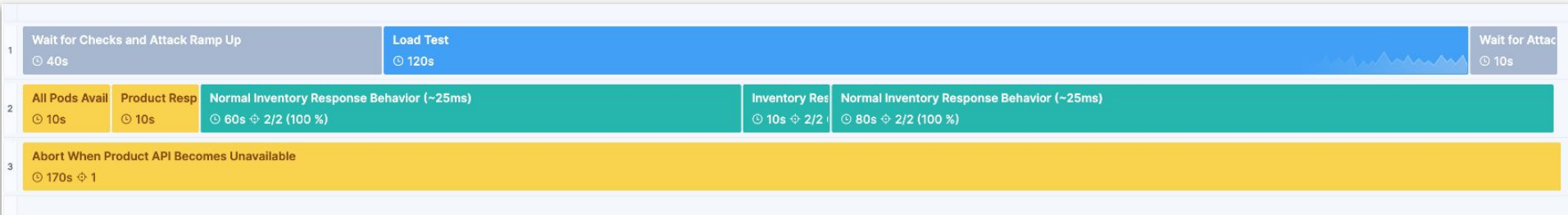


# Real Conditions

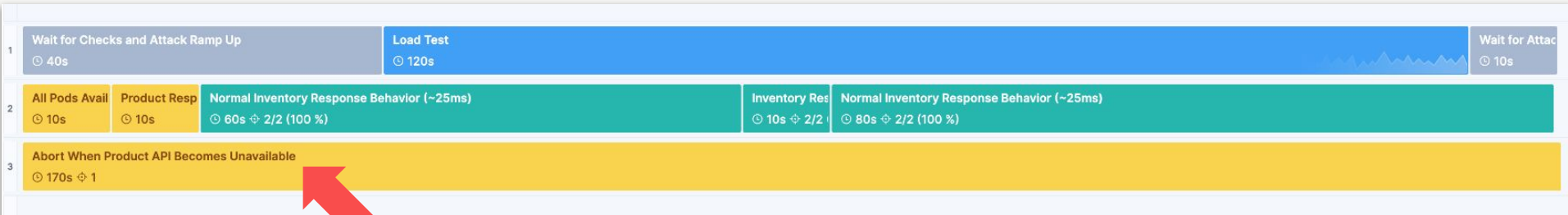
*Inventory* response time spikes up to 500ms



# Experiment Design



# Experiment Design



Conditions under which the experiment fails

# Experiment Run

#35661 Completed  
24/01/2023, 18:15:07 by nil.wloka

Attack Monitor Agent Log Action Logs

Wait for Checks and Attack Ramp Up Load Test Wait for Attack R...

All Product Response: Normal Inventory Response Behavior (~20ms) Inventory Response: Normal Inventory Response Behavior (~20ms)

Abort When Product API Becomes Unavailable

Run Log

- 18:15:07 ✓ Experiment execution created by nil.wloka.
- 18:15:07 ✓ Establishing connection to agents...
- 18:15:14 ✓ Wait for Checks and Attack Ramp Up (Wait 40s)
- 18:15:14 ✓ **Check Available** succeeded after 3 seconds.
- 18:15:14 ✓ **Abort When Product API Becomes Unavailable** on Datadog monitor **Synthetic Shopping Post API Availability** succeeded after 17s seconds.
- 18:15:18 ✓ **Product Response Available** succeeded after 18 seconds.
- 18:15:29 ✓ **Inventory Response Available (~20ms)** on JMW application **Inventory, InventoryApplication, nil\_inventory\_inventory\_0e4209b5-79ee8-4ad0-b8a4-1796b2934f43** completed.
- 18:15:29 ✓ **Normal Inventory Response Behavior (~20ms)** on JMW application **Inventory, InventoryApplication, nil\_inventory\_inventory\_0e4209b5-79ee8-4ad0-b8a4-1796b2934f43** completed.
- 18:15:56 ✓ **Load Test** completed. **RESTEST-0\_35661\_k6\_ter**
- 18:16:32 ✓ **Destroy Resource (no kube config)** completed on JMW application

Deployment Readiness

All 20 deployments have as many pods ready as desired.

Kubernetes Event Log

- 18:15:38 Normal Valid store validated
- 18:16:13 Normal Updated updated Secret
- 18:16:34 Normal Valid store validated
- 18:16:35 Warning Unhealthy Liveness probe failed: Get "http://10.1.83.70:8080/actuator/health/liveness": context deadline exceeded (...

HTTP Responses for http://demo.steadybit.io/products/circuitbreaker

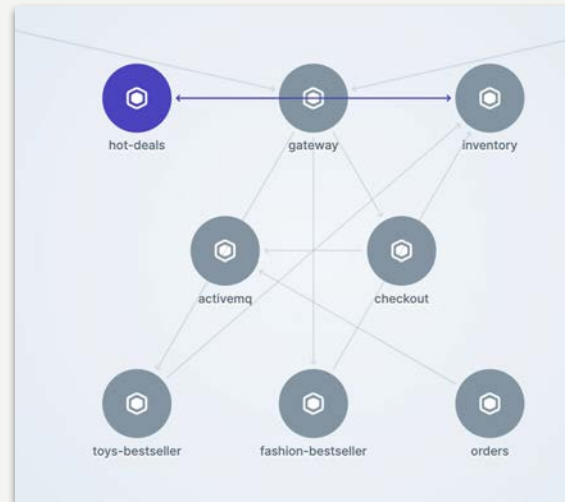
Datadog Monitor Status

Synthetic Shopping...



# Real Conditions

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



# Experiment Design

The screenshot displays the Steadybit experiment design interface. The main area is a timeline from 00:00 to 00:02:10. The experiment consists of three steps:

- Step 1:** HTTP (100s)
- Step 2:** Wait (10s) followed by Hot-Deals service delay in eu-central-1a & eu-central-1b (60s, 4/4 (100 %))
- Step 3:** Wait (60s) followed by DNS outage k8s cluster demo-prod (30s, 11/11 (100 %))

The right-hand panel is titled "HTTP GET" and contains the following configuration sections:

- Request Definition:**
  - HTTP Method: GET
  - Target URL: http://k8s.demo.steadybit.io/pro...
  - HTTP Headers: Key, Value
  - Additional Settings: >
- Repetition Control:**
  - How would you like to specify request repetitions?
    - Requests per second
    - Total number of Requests
  - Requests per second: 1
  - Duration: 100 Seconds
  - Additional Settings: >
- Result Verification:**
  - Response status codes: 200-299
  - Required Success Rate: 100 %

# Experiment Run

#35651

FAILED

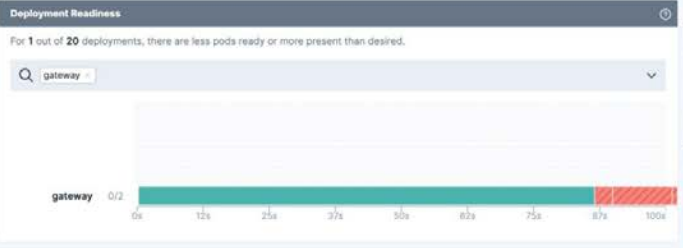
24/01/2023, 15:01:00 by Benjamin Wilms

Attack Monitor

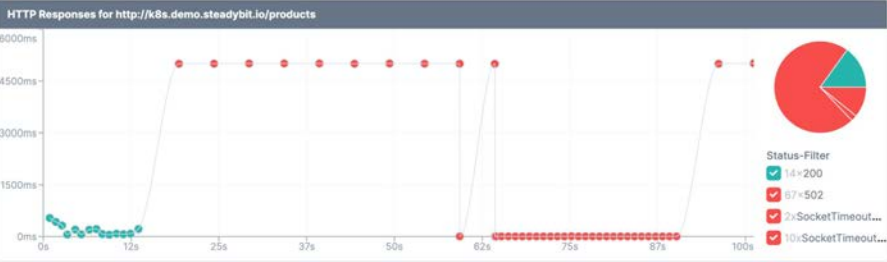
Agent Log



Run Log	
15:01:00	✓ Experiment execution created by Benjamin Wilms.
15:01:00	✓ Establishing connection to agents ...
2 15:01:13	✓ Waiting 10s
3 15:01:13	✓ Waiting 60s
3 15:01:14	⚠ HTTP check failed, Success-Rate is below expected. (15% < 100%)
2 15:01:32	✓ Hot-Deals service delay in eu-central-1a & eu-central-1b on container Hot-Deals k8s_hot-deals_hot-deals-5f66e558f-hqz28.steadybit-demo_ad-e9881-f52c-4290-9a26-4bf89b0c89aa_0 completed.
2 15:01:32	✓ Hot-Deals service delay in eu-central-1a & eu-central-1b on container Hot-Deals nginx_k8s_hot-deals-nginx_hot-deals-5f66e558f-hqz28.steadybit-demo_ad-e9881-f52c-4290-9a26-4bf89b0c89aa_0 completed.
2 15:01:36	✓ Hot-Deals service delay in eu-central-1a & eu-central-1b on container Hot-Deals nginx_k8s_hot-deals-nginx_hot-deals-5f66e558f-hqz28.steadybit-demo_405cd497-956d-4a2f-8d19-8c3eb0b0c027_0 completed.
2 15:01:37	✓ Hot-Deals service delay in eu-central-1a & eu-central-1b on container Hot-Deals nginx_k8s_hot-deals-nginx_hot-deals-5f66e558f-hqz28.steadybit-demo_405cd497-956d-4a2f-8d19-8c3eb0b0c027_0 completed.



Kubernetes Event Log			
15:01:19	Normal	Valid	store validated
15:01:32	Warning	Unhealthy	Readiness probe failed: Get "http://10.1.88.84:8080/actuator/health/readiness": -
15:01:42	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.88.84:8080/actuator/health/liveness": co.
15:01:55	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.83.70:8080/actuator/health/liveness": co.
15:01:52	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.88.84:8080/actuator/health/liveness": co.
15:01:52	Warning	Unhealthy	Readiness probe failed: Get "http://10.1.88.84:8080/actuator/health/readiness": -
15:02:05	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.83.70:8080/actuator/health/liveness": co.
15:02:02	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.88.84:8080/actuator/health/liveness": co.
15:02:02	Normal	Killing	Container gateway failed liveness probe, will be restarted
15:02:15	Warning	Unhealthy	Liveness probe failed: Get "http://10.1.83.70:8080/actuator/health/liveness": co.
15:02:15	Normal	Killing	Container gateway failed liveness probe, will be restarted
15:02:11	Warning	Unhealthy	Readiness probe failed: Get "http://10.1.88.84:8080/actuator/health/readiness": -



**When should we run these experiments?**

Make it an  
**integral** part of the software  
development process

# GitHub Action

The screenshot shows a GitHub Actions workflow run for the 'run\_experiment' job in the 'steadybit/shopping-demo' repository. The workflow is titled 'Add resilience test action #8' and is part of a 'Resilience Test' workflow. The run is successful and completed 3m 2s ago.

**Jobs:**

- run\_experiment

**Run details:**

- Usage
- Workflow file

**run\_experiment**  
succeeded last year in 3m 2s

**Set up job** 1s

```
1 Current runner version: '2.299.1'
2 Operating System
3 Runner Image
4 Runner Image Provisioner
5 GITHUB_TOKEN Permissions
6 Secret source: Actions
7 Prepare workflow directory
8 Prepare all required actions
9 Getting action download info
10 Download action 'repository: 'steadybit/run-experiment@v1' (SHA:18ec29eb4c684457329602fa85948c2ad4e695ce)
```

**Run Resilience Regression Test in Steadybit** 3m 0s

```
1 Run steadybit/run-experiment@v1
2 with:
3   accessToken: **
4   experimentKey: RESTEST-9
5   baseUrl: https://platform.steadybit.io
6   expectedState: COMPLETED
7   maxRetries: 3
8   maxRetriesOnExpectationFailure: 0
9   delayBetweenRetriesOnExpectationFailure: 60000
10 Triggering experiment RESTEST-9 for attempt 1/1.
11 Experiment RESTEST-9 is running, checking status...
12 Experiment RESTEST-9 ended. Execution 35198 ended with 'COMPLETED'.
```

**Complete job** 0s

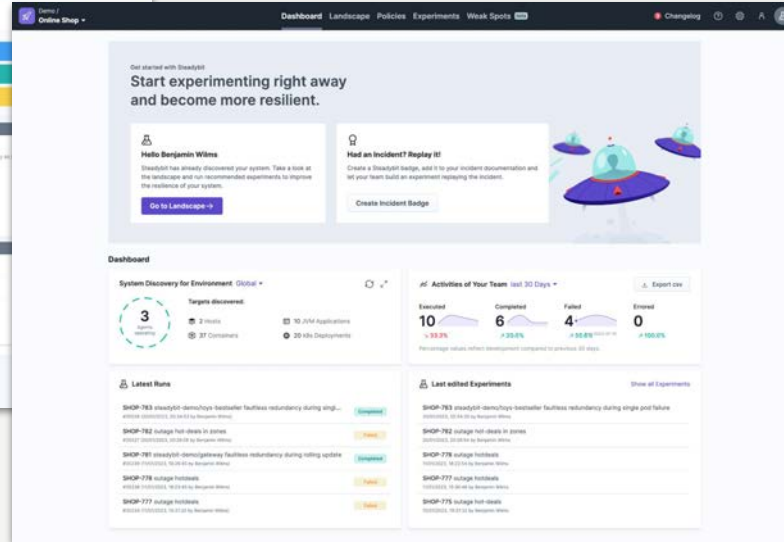
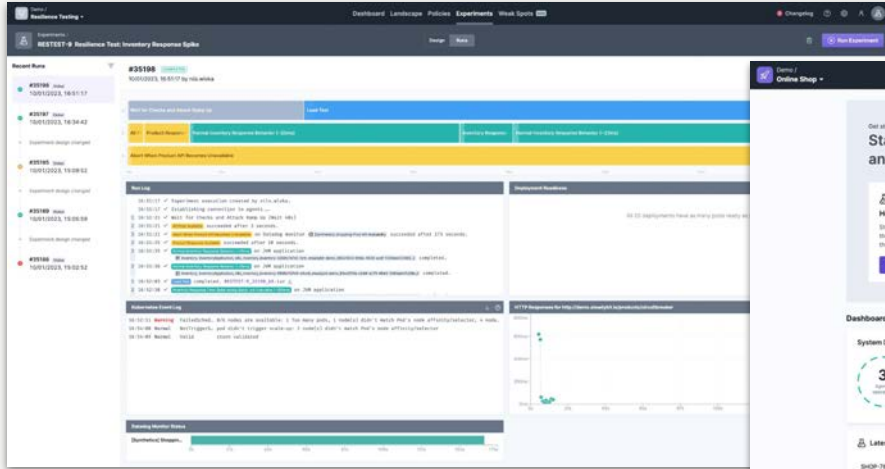
```
1 Cleaning up orphan processes
```

# Recap

**Embrace failures and turn them into  
experiments**



# Chaos Engineering with Steadybit



# Thank you

