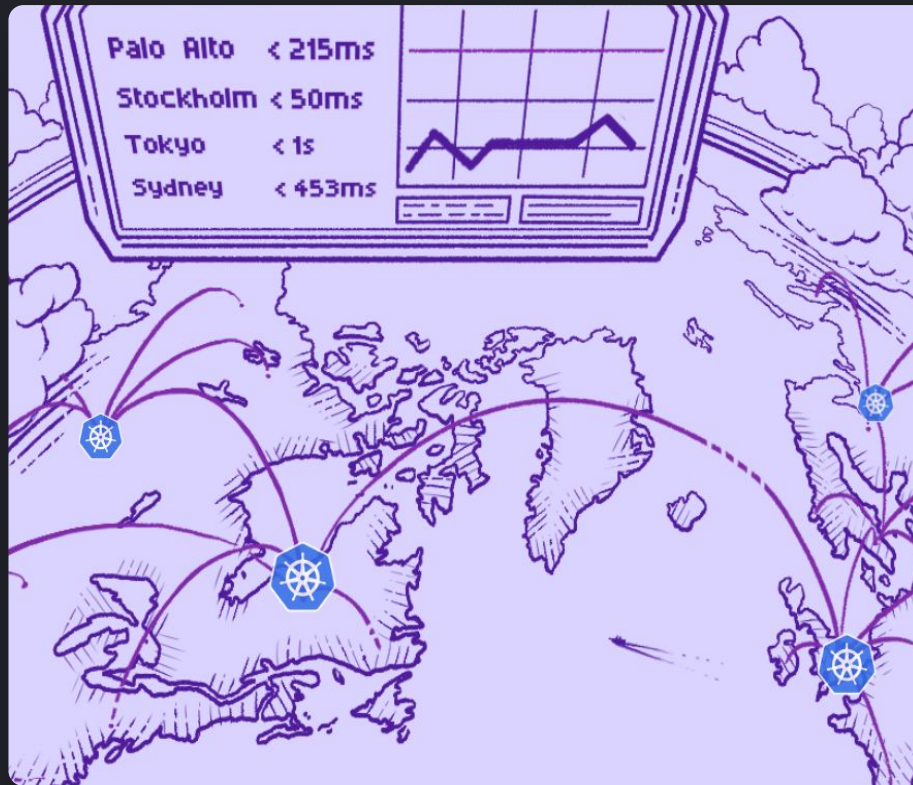


# Running distributed load tests with k6



**Paul Balogh**

Developer Advocate, k6 OSS  
@javaducky



“ ”

By bringing k6 and Grafana together, we enable teams to radically improve the time to **prevent, detect, and remediate problems** before they impact end users.



Raj Dutt

Grafana Labs CEO/Co-founder





# Overview

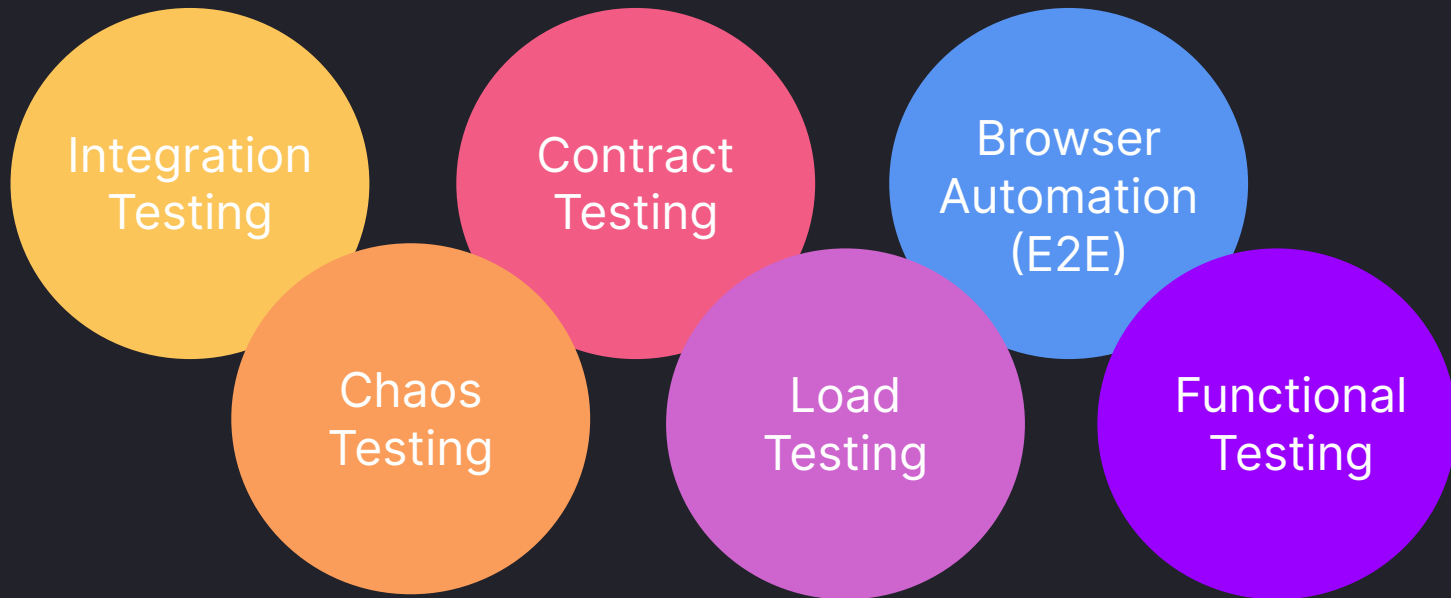
- 1 **What is k6?**
- 2 What is load testing?
- 3 Introducing the k6-operator
- 4 Where do we go from here?



A **reliability testing** tool.



# k6: a reliability testing tool



“ ”

Although it's easy to get started with k6 [OSS] all on its own, it really shines with its ease of integration into a developer ecosystem.



Thoughtworks Technology Radar  
April 2021



# k6: a reliability testing tool

## OpenSource

OSS is at the heart of **what we do** and helps leave the world a little better than we found it

## Scriptable

CLI and API designed for **automating** your tests with pass/fail criteria using JavaScript syntax

## Performant

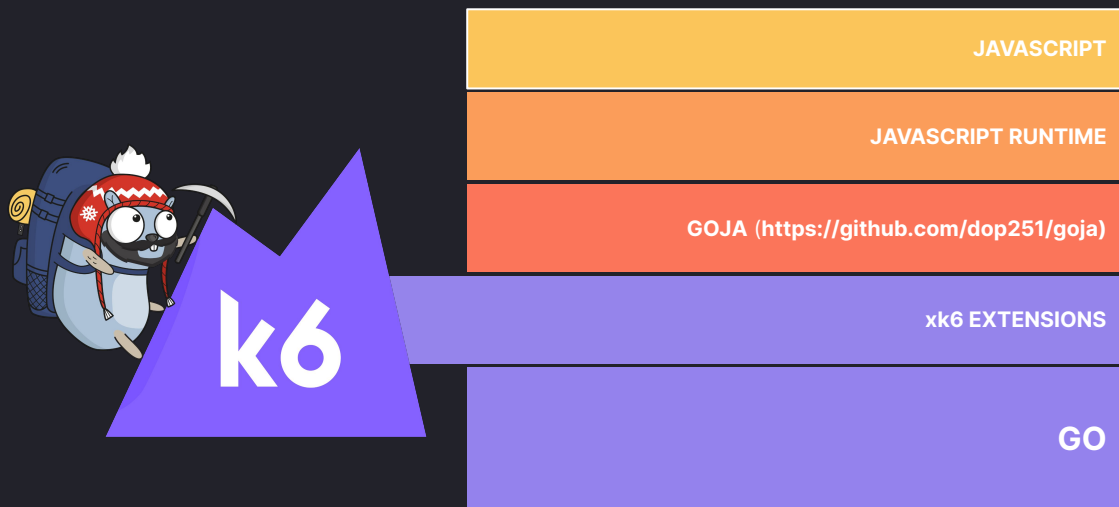
A k6 engine **written in Go** making it one of the the best performing tools available

## Extensible

Use Go(lang) code to **add support** for new outputs, protocols, and products from within your test scripts



# k6: a reliability testing tool







# Overview

- 1 What is k6?
- 2 **What is load testing?**
- 3 Introducing the k6-operator
- 4 Where do we go from here?



“ ”

Load testing is the process of putting **demand** on a system and **measuring** its **response**.



The Internet  
Knower of all things



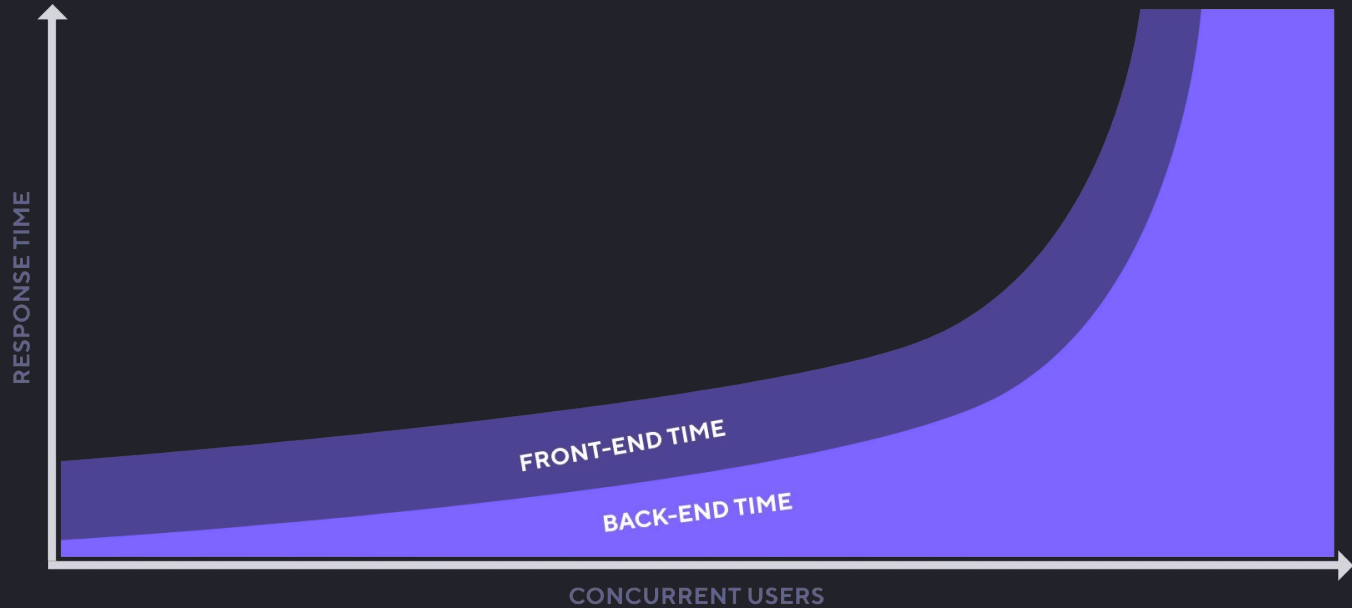
# Myths about load testing

- Performance testing == load testing
- It is only for **large** companies
- Is **expensive** to do
- Should **only** be in production
- Unnecessary if you have o11y



# Why do load testing today?

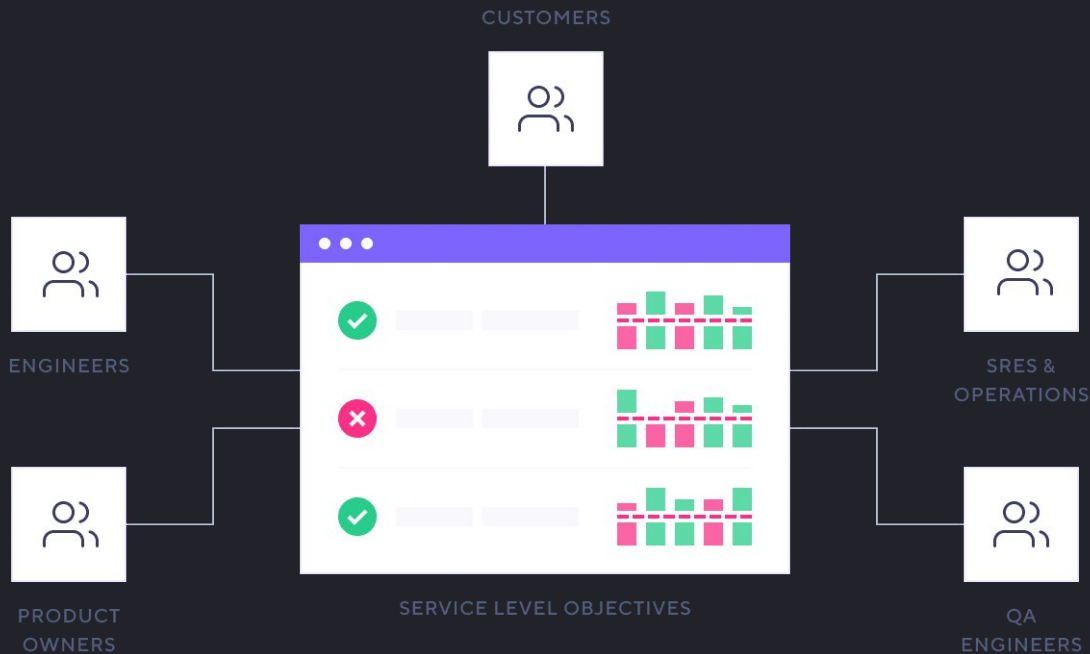
UX



# Why do load testing today?

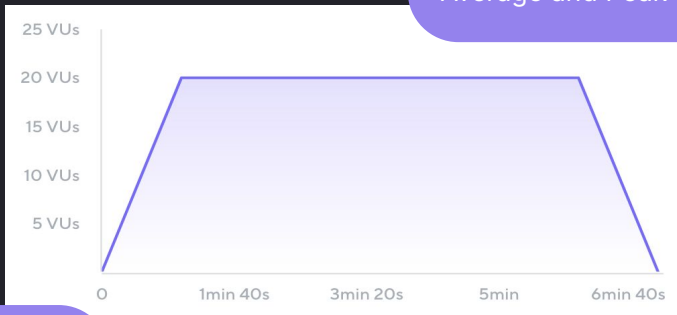
Proactively test

# SLOs

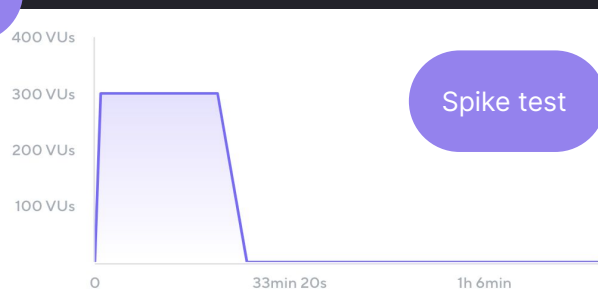


# Common types of load tests

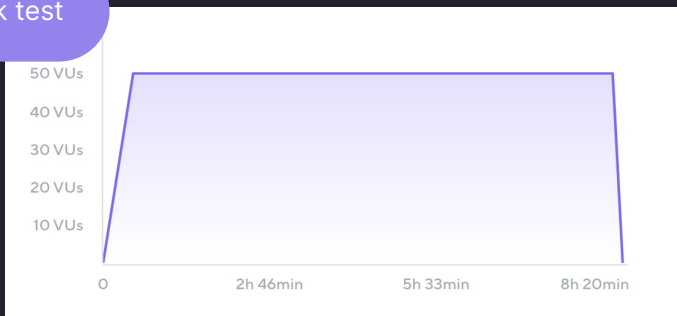
Average and Peak test



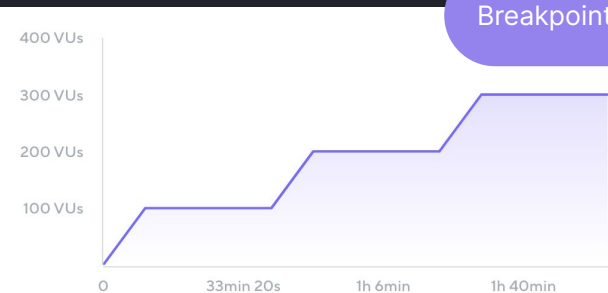
Spike test



Soak test



Breakpoint test



# Modeling your load tests

Shared iterations

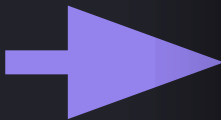
Per VU iterations

Constant VUs

Ramping VUs

Constant Arrival Rate

Ramping Arrival Rate



```
export const options = {
  scenarios: {
    mainScenario: {
      executor: 'constant-vus',
      exec: 'myUserFlowA',
      vus: 10,
      duration: '60m',
    },
    scenarioB: {
      executor: 'shared-iterations',
      exec: 'myUserFlowB',
      startTime: '30m',
    },
  },
};
```





# Overview

- 1 What is k6?
- 2 What is load testing?
- 3 **Introducing the k6-operator**
- 4 Where do we go from here?





# Custom Resource Definition (CRD) launcher

Same k6 test script, multiple execution modes!

```
>_ k6 run script.js LOCAL
```

```
>_ kubectl create cm test-scripts K8S
```

```
>_ kubectl apply -f k6.yaml K8S
```

```
>_ k6 cloud script.js CLOUD
```





US WEST



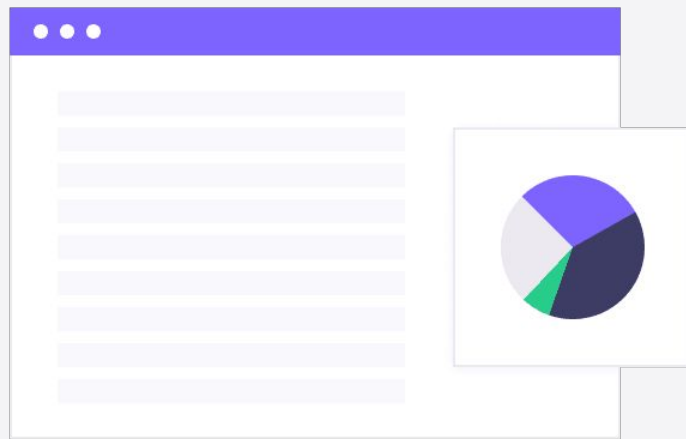
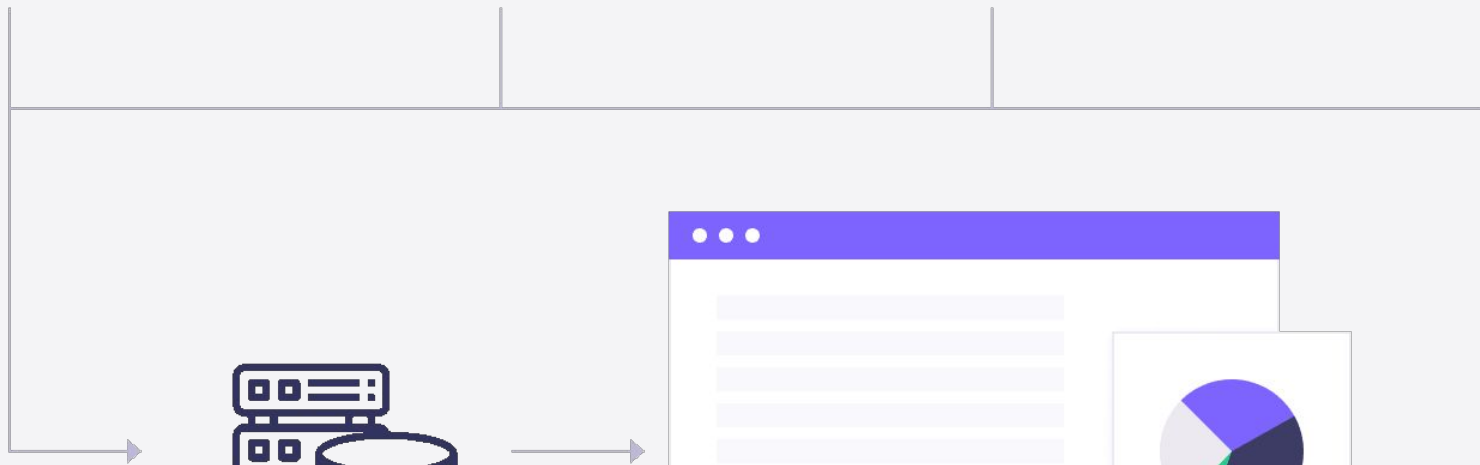
US EAST



CENTRAL EU



ASIA

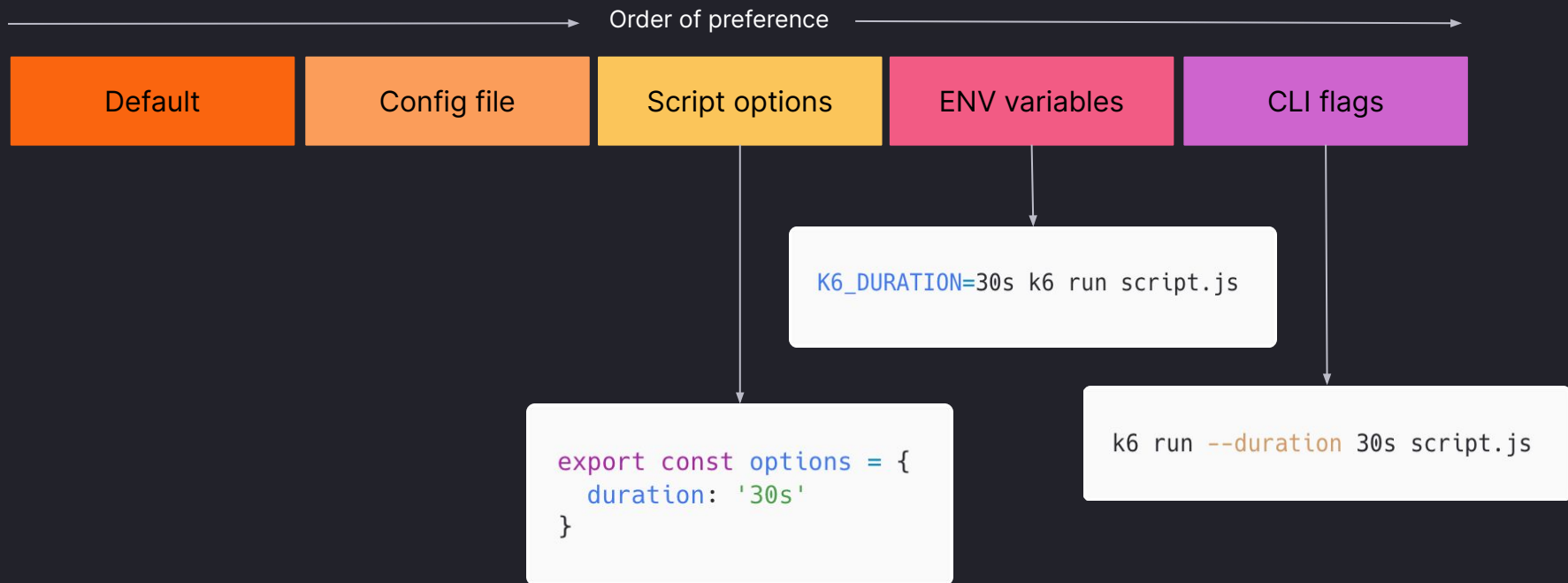


VISUALIZATION



# Worker nodes

57 configurable options → <https://k6.io/docs/using-k6/k6-options/reference/>



# Demo!





# Overview

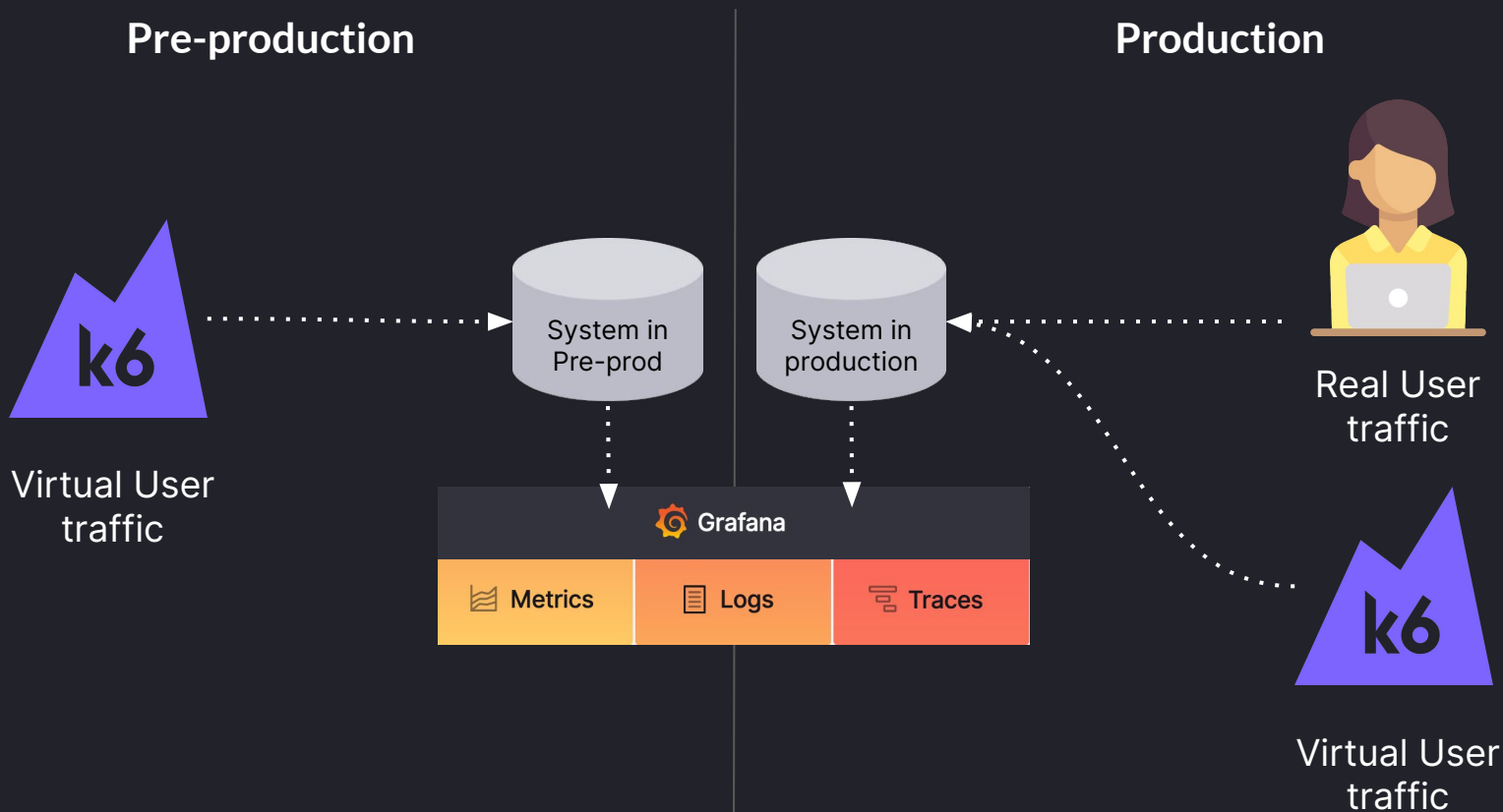
- 1 What is k6?
- 2 What is load testing?
- 3 Introducing the k6-operator
- 4 **Where do we go from here?**



# The “Golden Signals”



# Proactively improve reliability



# Thanks for participating!

Connect with Paul as

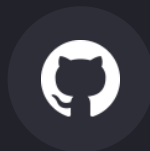
[@javaducky](#) or [linkedin/in/pabalogh](#)

Demonstration project available at

[github.com/javaducky/demo-k6-operator](https://github.com/javaducky/demo-k6-operator)



[k6.io/slack](#)



[grafana/k6](#)



[community.k6.io](#)