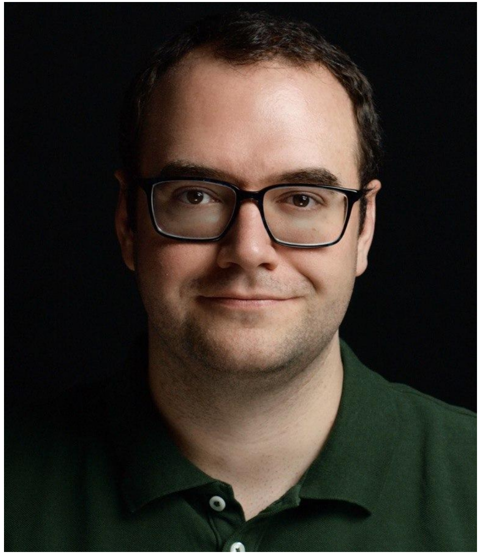
Progressive Delivery with Flux & Flagger

Max Körbächer - Kubernetes & Cloud Native Advocate

REPLY
STORM

Speaker

Max Körbächer



Kubernetes & Cloud Native Advocate

Working @ Storm Reply

K8s Release Team Comms Shadow

Get connected:

- Linkedin /maxkoerbaecher
- Twitter /mkoerbi
- Github /mkorbi

CERTIFIED kubernetes ADMINISTRATOR

CERTIFIED kubernetes APPLICATION DEVELOPER

kubernetes CERTIFIED SERVICE PROVIDER

kubernetes TRAINING PARTNER

Where are we now?

CI/CD, DevOps, Agile and Co.

Agile

focused on
process

highlighting
change

accelerating
delivery

CI/CD

focused on
software life-cycle

highlighting
tools

accelerating
automation

DevOps

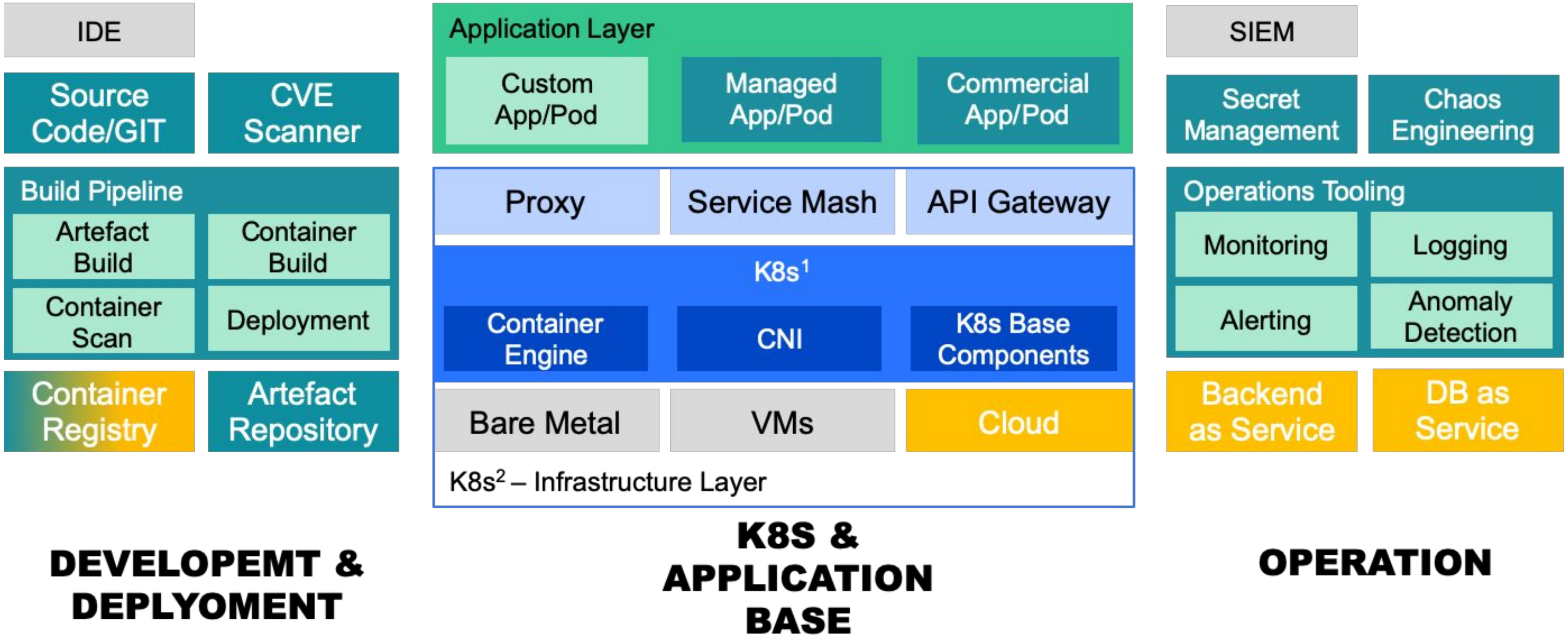
focused on
culture

highlighting
roles

accelerating
responsiveness

icons by Eucaly/happy

The complexity of full/high automation



IDE

Source Code/GIT

CVE Scanner

Build Pipeline

Artefact Build

Container Build

Container Scan

Deployment

Container Registry

Artefact Repository

DEVELOPEMT & DEPLYOMENT

Application Layer

Custom App/Pod

Managed App/Pod

Commercial App/Pod

Proxy

Service Mash

API Gateway

K8s[1]

Container Engine

CNI

K8s Base Components

Bare Metal

VMs

Cloud

K8s[2] – Infrastructure Layer

K8S & APPLICATION BASE

SIEM

Secret Management

Chaos Engineering

Operations Tooling

Monitoring

Logging

Alerting

Anomaly Detection

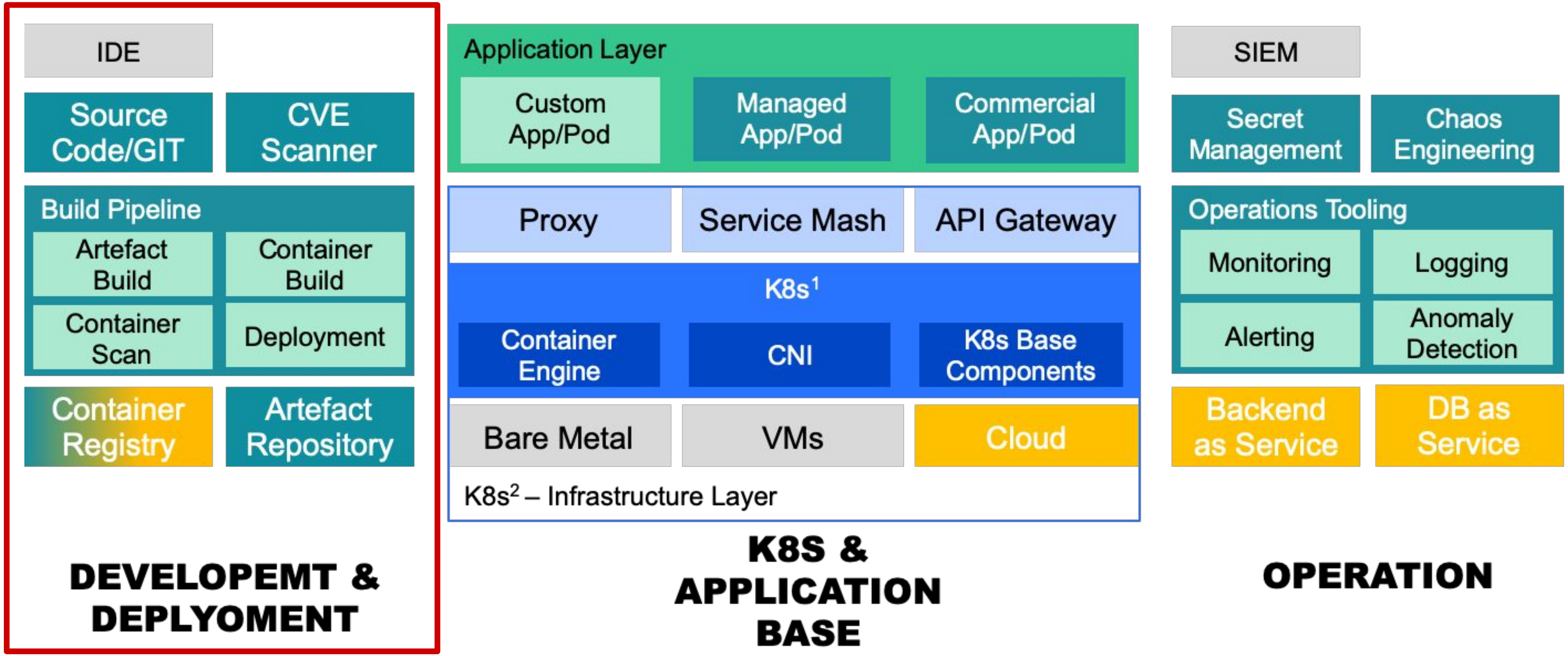
Backend as Service

DB as Service

OPERATION

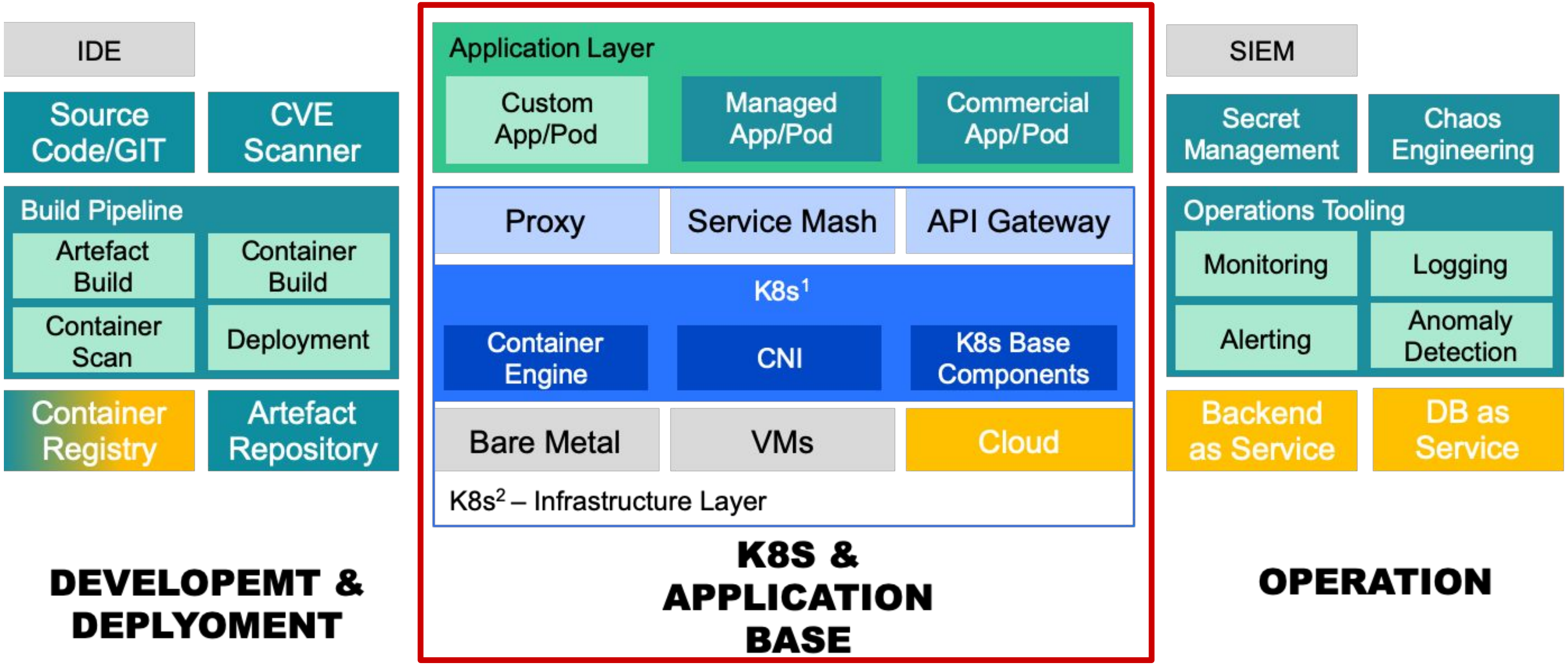
The complexity of full/high automation

Where we want to focus on



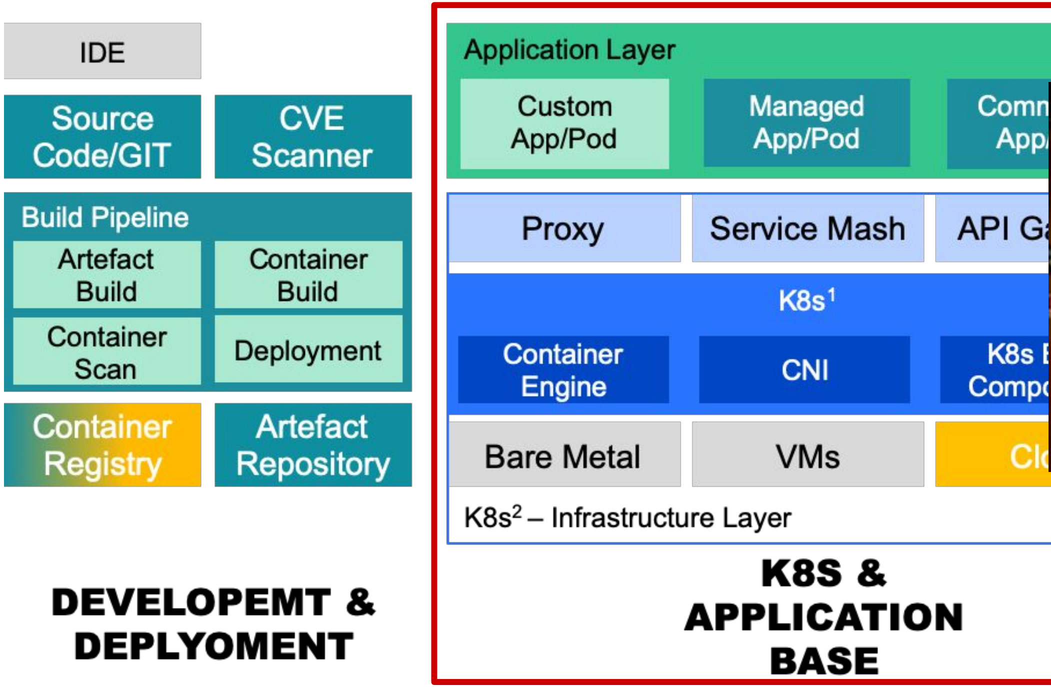
The complexity of full/high automation

Where we have an impact on



The complexity of full/high automation

Where we have an impact on



IDE

Source Code/GIT

CVE Scanner

Build Pipeline

Artefact Build

Container Build

Container Scan

Deployment

Container Registry

Artefact Repository

DEVELOPEMT & DEPLYOMENT

Application Layer

Custom App/Pod

Managed App/Pod

Comm App/

Proxy

Service Mash

API Ga

K8s^1

Container Engine

CNI

K8s E Compo

Bare Metal

VMs

Cl

K8s^2 – Infrastructure Layer

K8S & APPLICATION BASE

SIEM

OPERATION

The complexity of full/high automation

The blast radius

IDE

Source Code/GIT

CVE Scanner

Build Pipeline

Artefact Build

Container Build

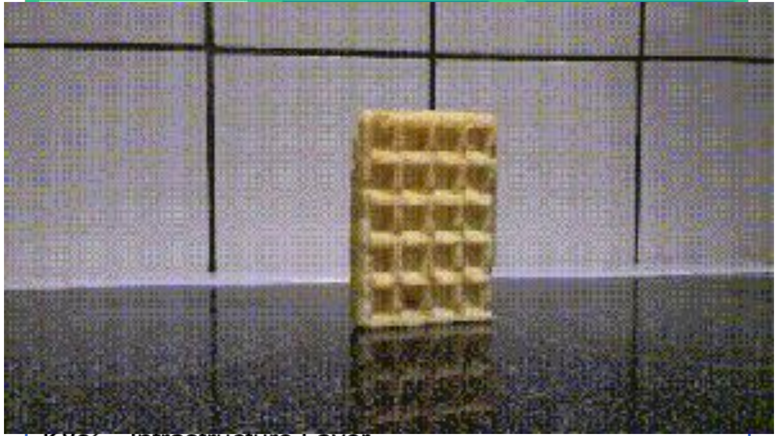
Container Scan

Deployment

Container Registry

Artefact Repository

Application Layer



K8s^2 – Infrastructure Layer

SIEM

Secret Management

Chaos Engineering

Operations Tooling

Monitoring

Logging

Alerting

Anomaly Detection

Backend as Service

DB as Service

DEVELOPEMT & DEPLYOMENT

K8S & APPLICATION BASE

OPERATION

What is the problem?

Rolling Updates

1 The de facto standard of release strategies. Gives you no/less control of the rollout speed.

No way back

2 Once started, a rollout is hard to stop and even harder to rollback. This requires in nearly all cases several major manual steps.

Liveness/Readiness Probes

3 Liveness and readiness probes are best practice to use and helpful for the lifecycle management from K8s perspective. But they don’t deliver deep insights or can use external metrics.

“Progressive delivery is continuous delivery with fine-grained control over the blast radius.”

— James Governor, RedMonk

Photo by Jeremy Perkins on Unsplash

Progressive Delivery

The next level of deployments

Progressive delivery requires high automation and rollback capabilities

Key element, and mostly identified with traffic shifting between different versions of the same application

It utilizes controlled methods like canary or blue/green deployment

Most powerful way to use: feature toggling - enable, disable or hide features

Progressive Delivery

The next level of deployments

Progressive deliver is often called “advanced deployment”, as it requires high automation

Key element, and mostly identified with traffic shifting between different versions of the same application

mainly implemented by

It utilizes controlled methods like canary or blue/green deployment

Most powerful way to use: feature toggling - enable, disable or hide features

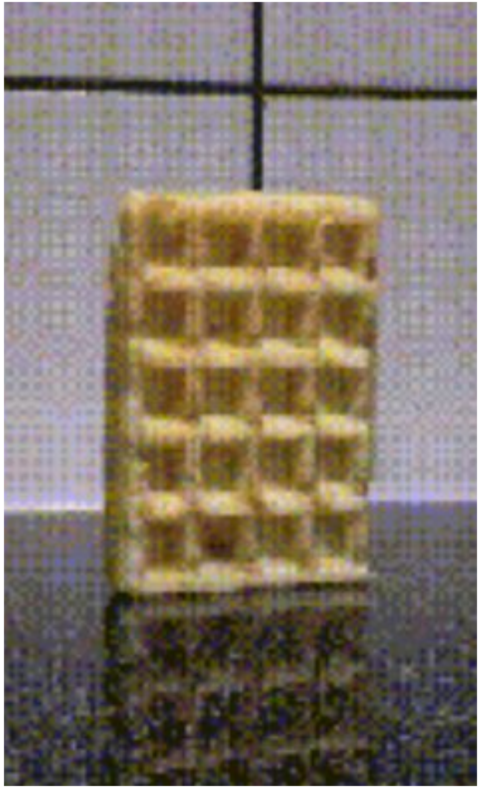
Service Meshes

- Actively managing the traffic within a K8s cluster, allows isolation, tracing and shifting traffic

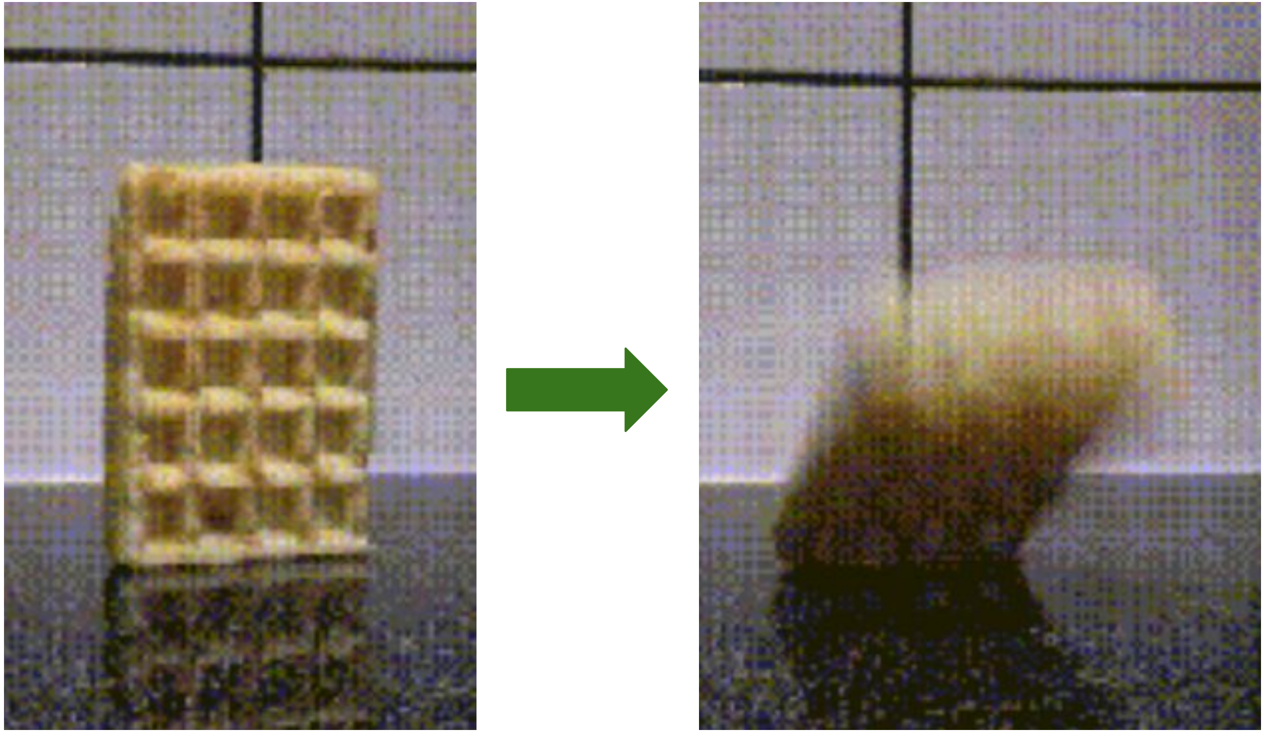
Observability

- Combining logs, metrics and tracing for deep insights of an application, function or component

In other words

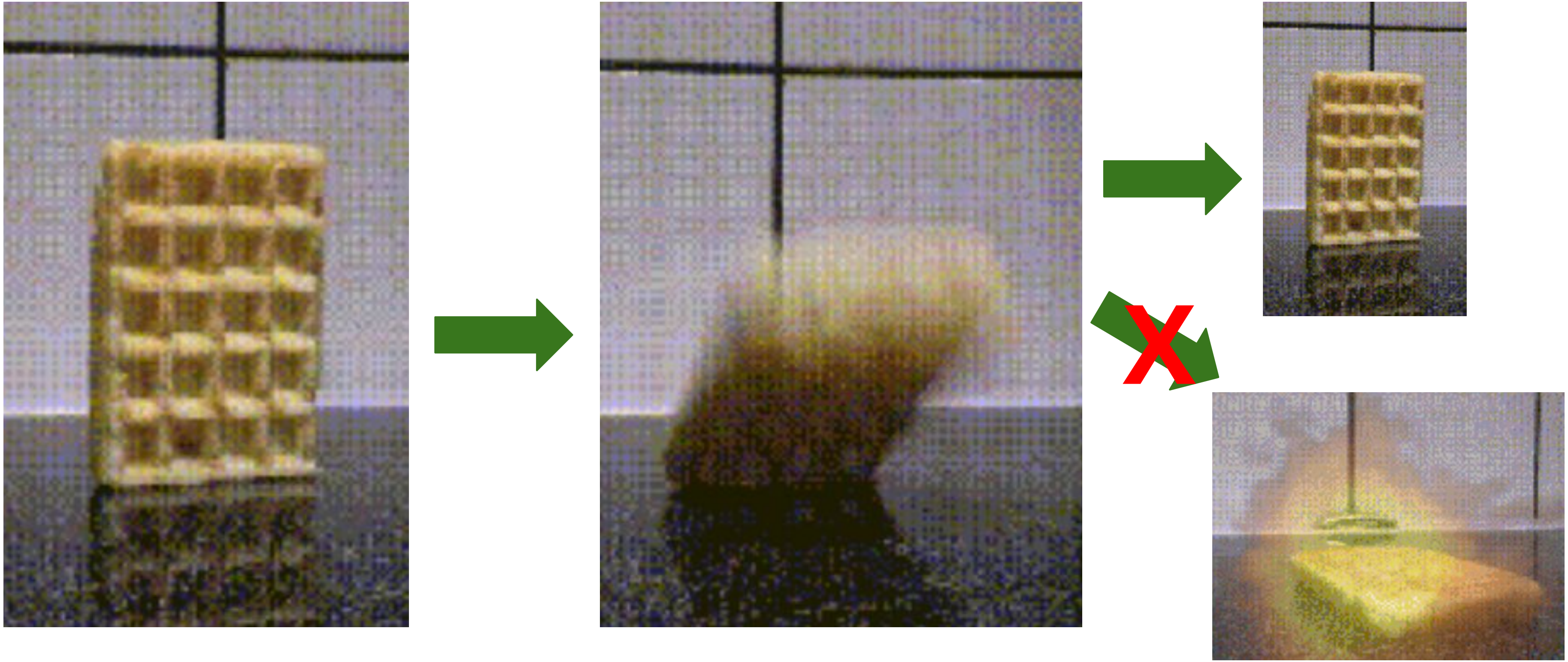


In other ~~words~~ images



In other ~~words~~ images

Progressive Delivery!



What can I do with with PD?

And how does it effect me?

I “pay” for:

- auto. rollbacks with failing updates
- be precises when an update is failing and when it is successful
- utilize extern, in depth metrics
- using different versions or configurations of an application

What can I do with with PD?

And how does it affect me?

I “pay” for:

- auto. rollbacks with failing updates
- be precises when an update is failing and when it is successful
- utilize extern, in depth metrics
- using different versions or configurations of an application

I get:

- end-to-end deployment of an application (and this time truly!)
- reducing/eliminating blast radius
- GitOps in heart, no pipeline dependencies
- Pure control based on metrics and configurations

Flux & Flagger

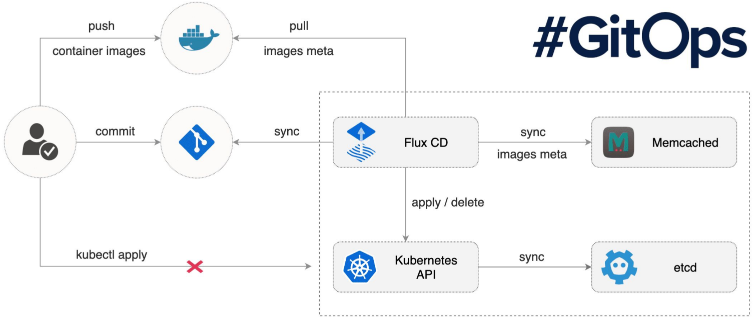
GitOps - Progressive Delivery

Flux

Declarative description in Git

Automate Deployment from a cluster not a pipeline

Push and sync code



Flux & Flagger

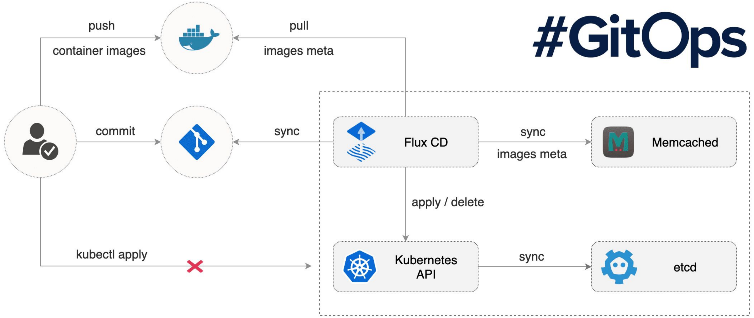
GitOps - Progressive Delivery

Flux

Declarative description in Git

Automate Deployment from a cluster not a pipeline

Push and sync code

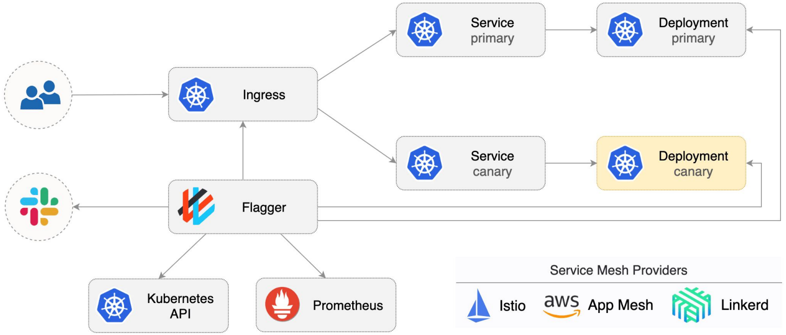


Flagger

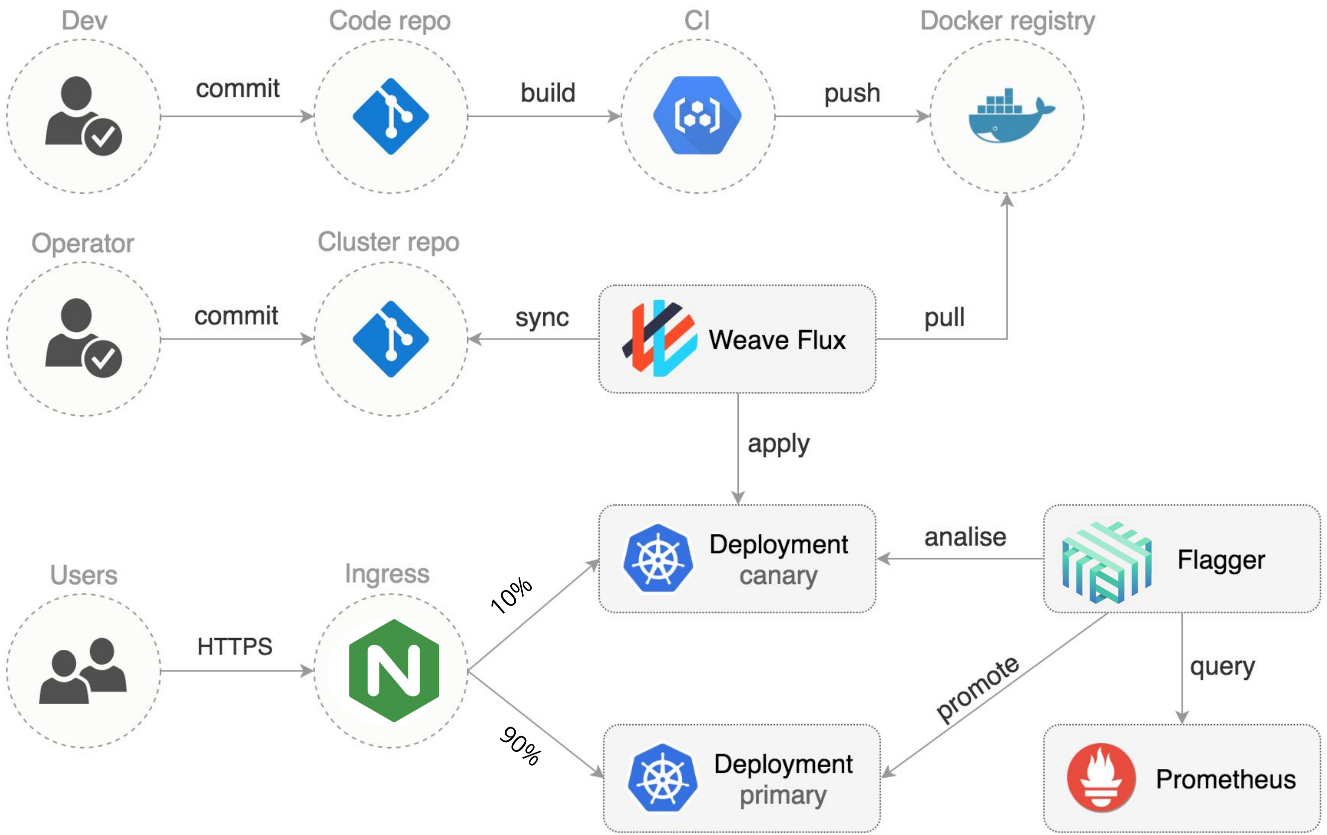
Implements deployment strategy for automatic release process

Can query metrics to gather feedback

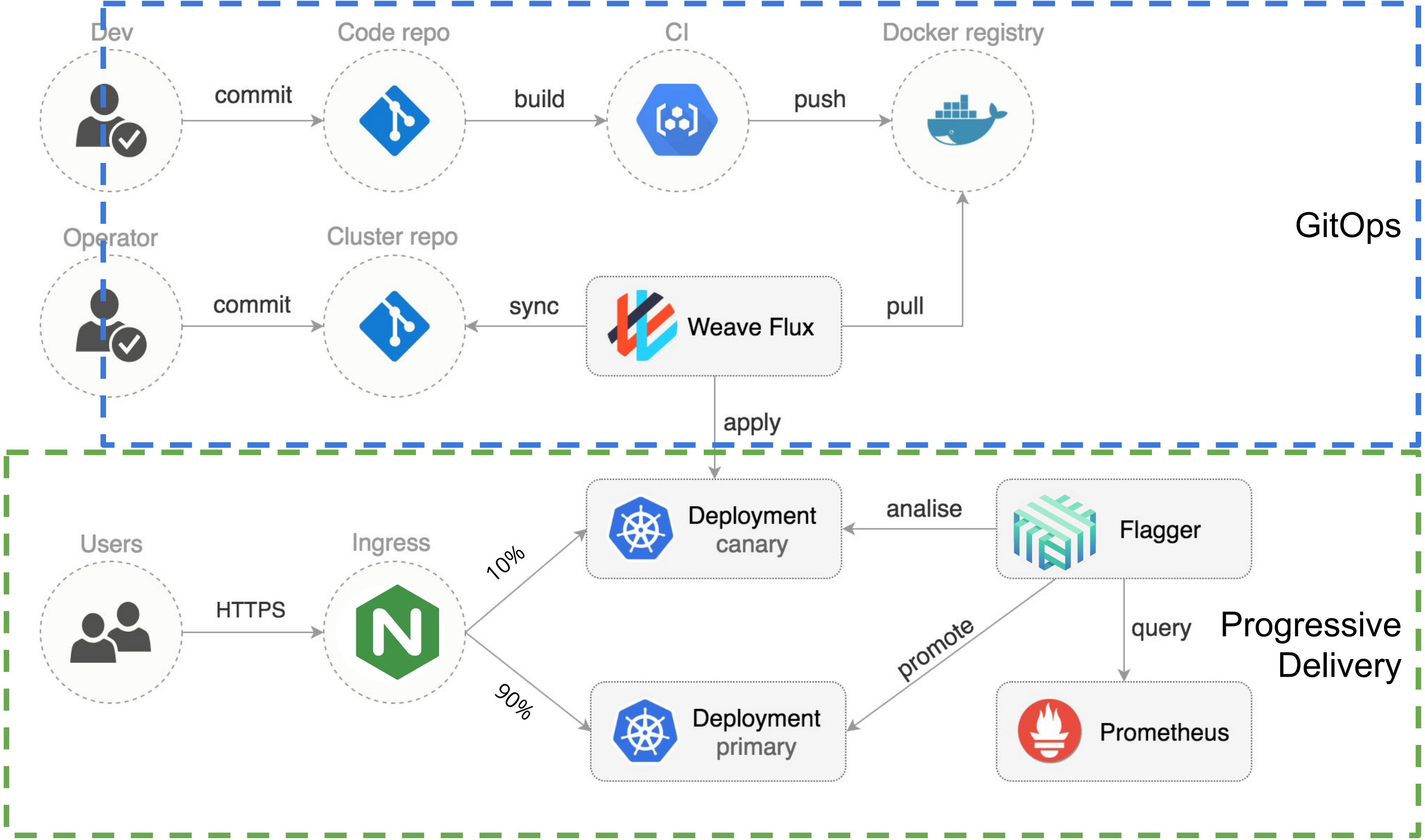
Run tests for conformance



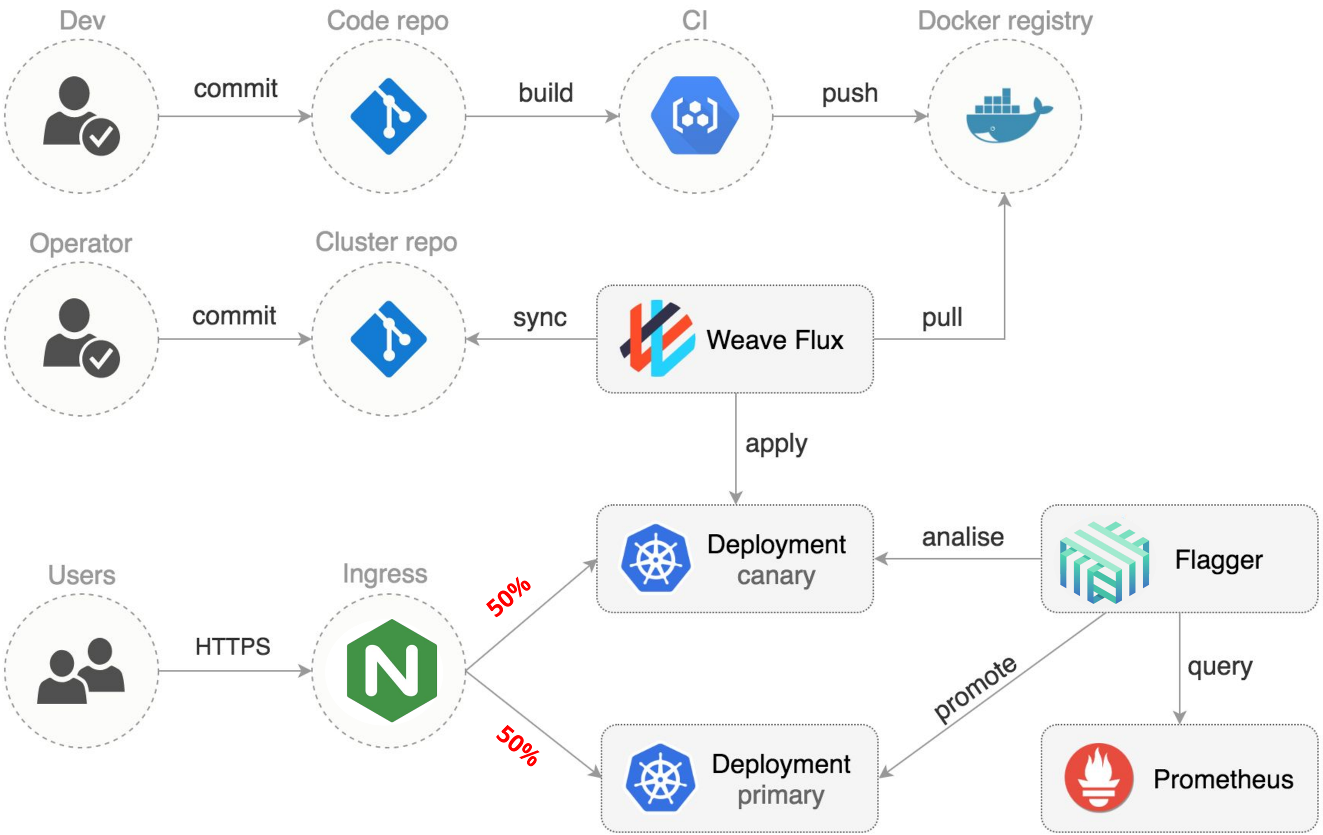
How a Progressive Delivery can looks like



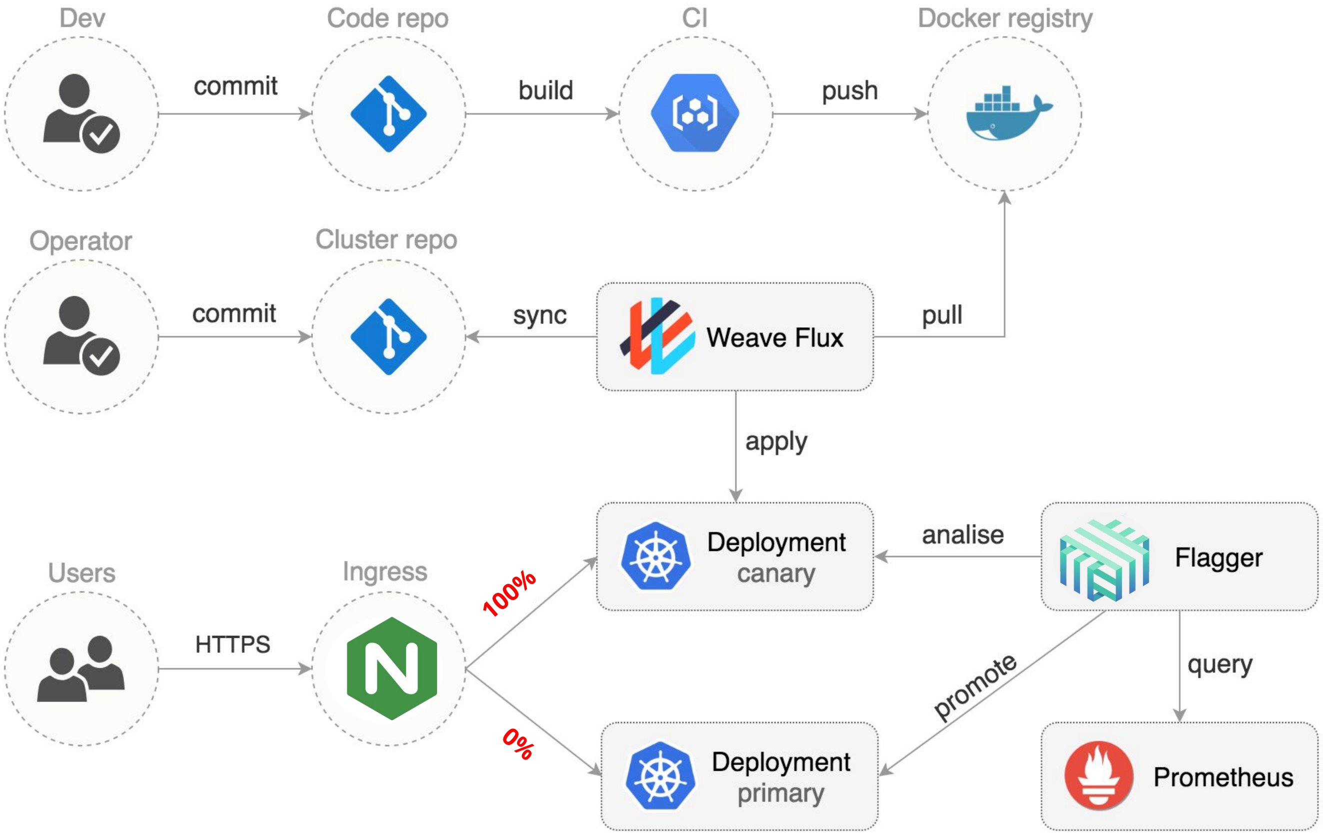
How a Progressive Delivery can looks like



How a Progressive Delivery can looks like



How a Progressive Delivery can looks like



Enough slides, let’s have a look!

Simple Release

Progressive Delivery Release

Let’s stay in touch!

maxkoerbaecher

mkorbi

mkoerbi

icons by Freepik