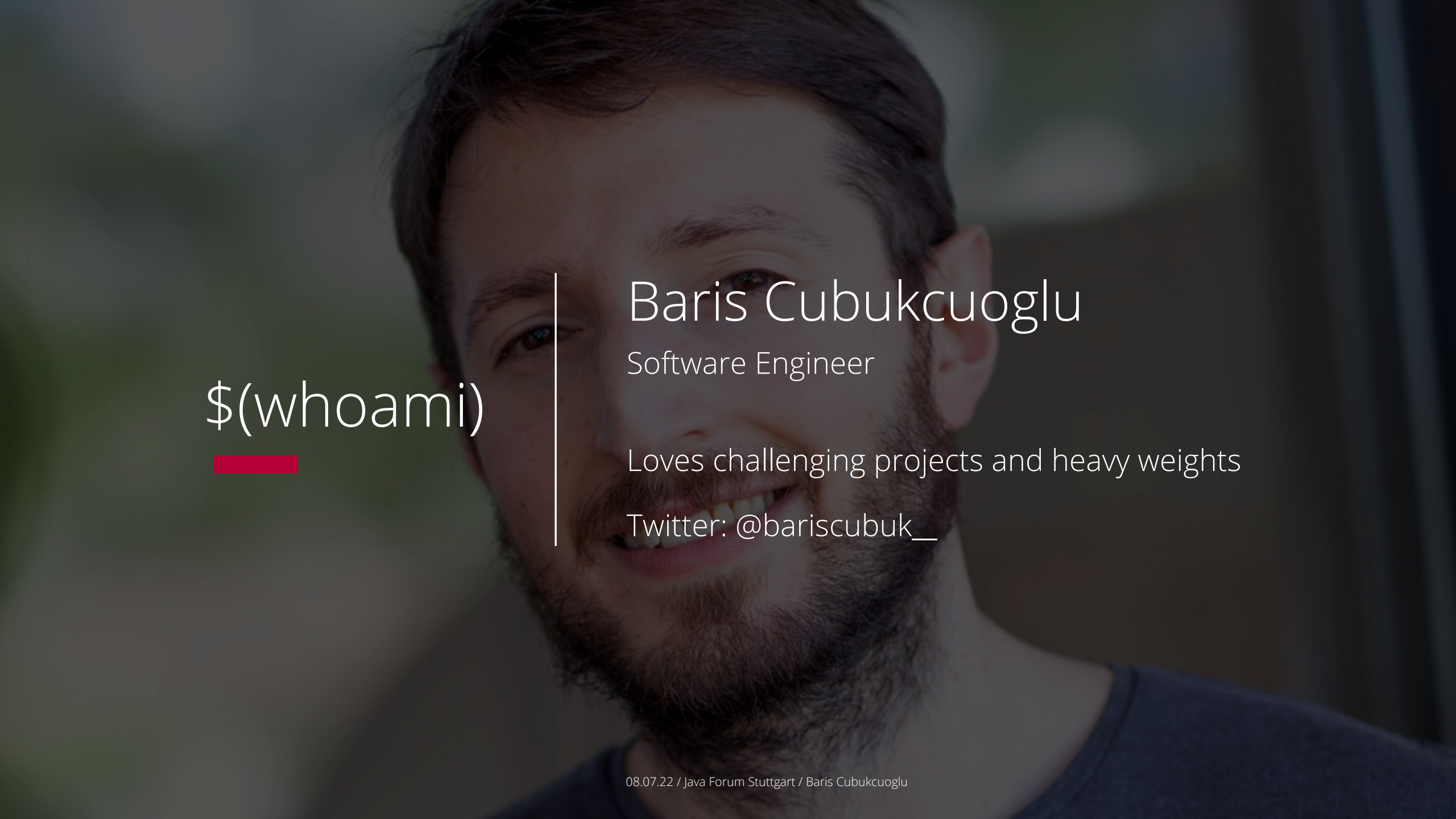
Ein ❤️ für Entwickler

Die Developer Experience in Kubernetes verbessern

// Baris Cubukcuoglu

MIMACOM



$(whoami)

Baris Cubukcuoglu

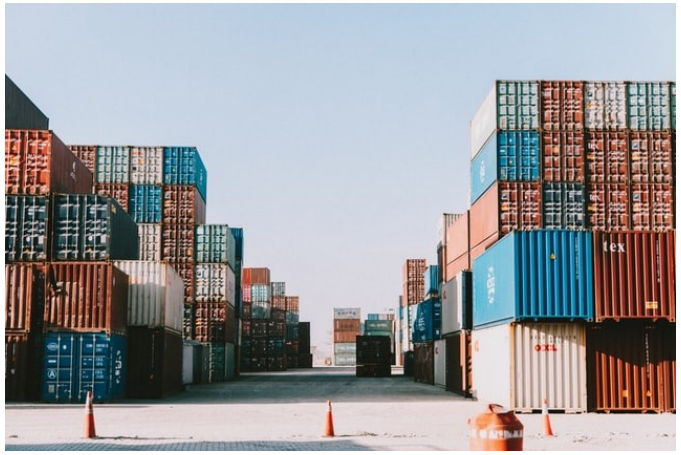
Software Engineer

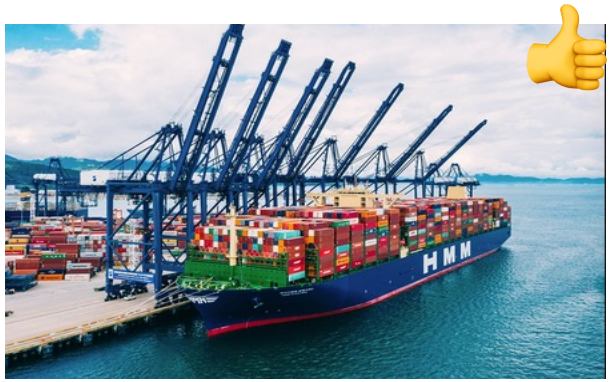
Loves challenging projects and heavy weights

Twitter: @bariscubuk__


MIMACOM

Developer Experience





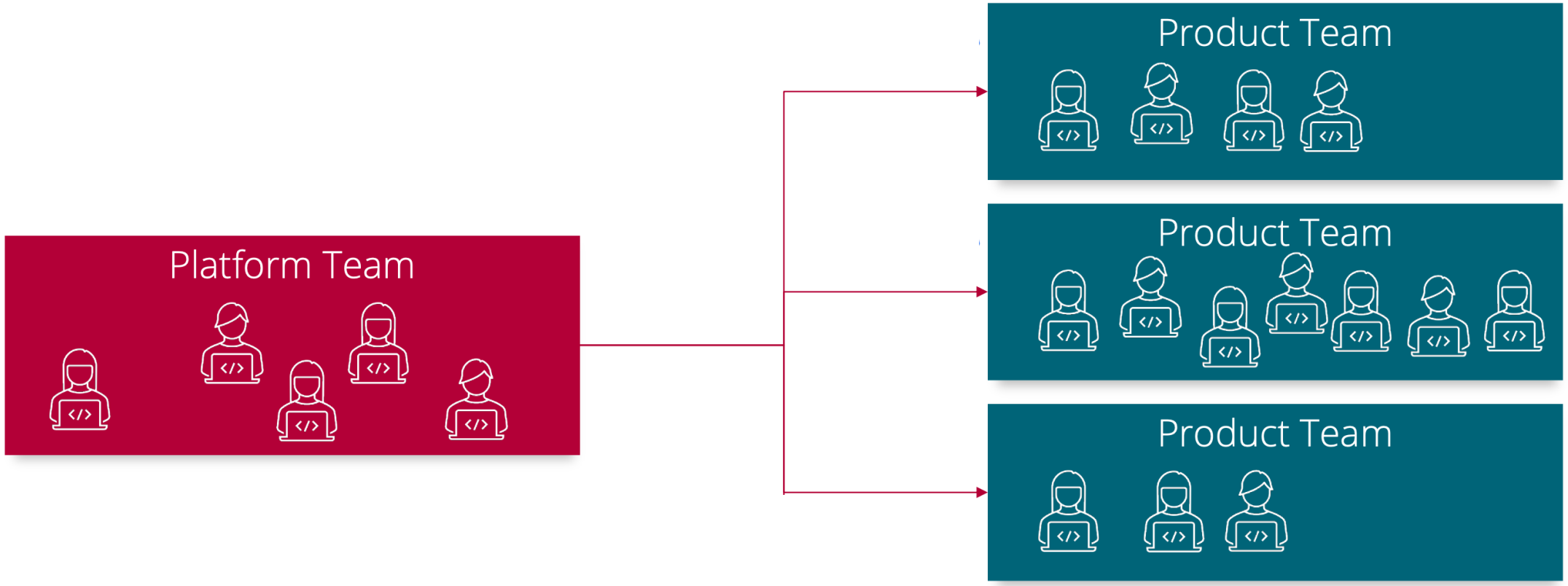
Responsibilities

CODING	DEPLOYMENT	PACKAGING	COMPUTE	ROUTING	OPERATIONS	SECURITY
App	CI/CD	Image Building	Network Setup	Ingress	Monitoring	Certificates In-Transit
	Deployment Strategy	Hardening	Compute Orchestration	Service Discovery	Log Aggregation	Encryption At-Rest
		OS	Infrastructure as Code		Backup & Recovery	Dev Access Control
						Security Concept

Business Value Impact

CODING	DEPLOYMENT	PACKAGING	COMPUTE	ROUTING	OPERATIONS	SECURITY
App	CI/CD	Image Building	Network Setup	Ingress	Monitoring	Certificates In-Transit
	Deployment Strategy	Hardening	Compute Orchestration	Service Discovery	Log Aggregation	Encryption At-Rest
		OS	Infrastructure as Code		Backup & Recovery	Dev Access Control
						Security Concept

Separation Is Key



Product Team

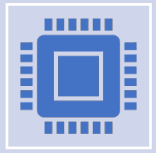
CODING	DEPLOYMENT	PACKAGING	COMPUTE	ROUTING	OPERATIONS	SECURITY
App	CI/CD	Image Building	Network Setup	Ingress	Monitoring	Certificates In-Transit
	Deployment Strategy	Hardening	Compute Orchestration	Service Discovery	Log Aggregation	Encryption At-Rest
		OS	Infrastructure as Code		Backup & Recovery	Dev Access Control
						Security Concept

Platform Team

CODING	DEPLOYMENT	PACKAGING	COMPUTE	ROUTING	OPERATIONS	SECURITY
App	CI/CD	Image Building	Network Setup	Ingress	Monitoring	Certificates In-Transit
	Deployment Strategy	Hardening	Compute Orchestration	Service Discovery	Log Aggregation	Encryption At-Rest
		OS	Infrastructure as Code		Backup & Recovery	Dev Access Control
						Security Concept

But why is Kubernetes so difficult?

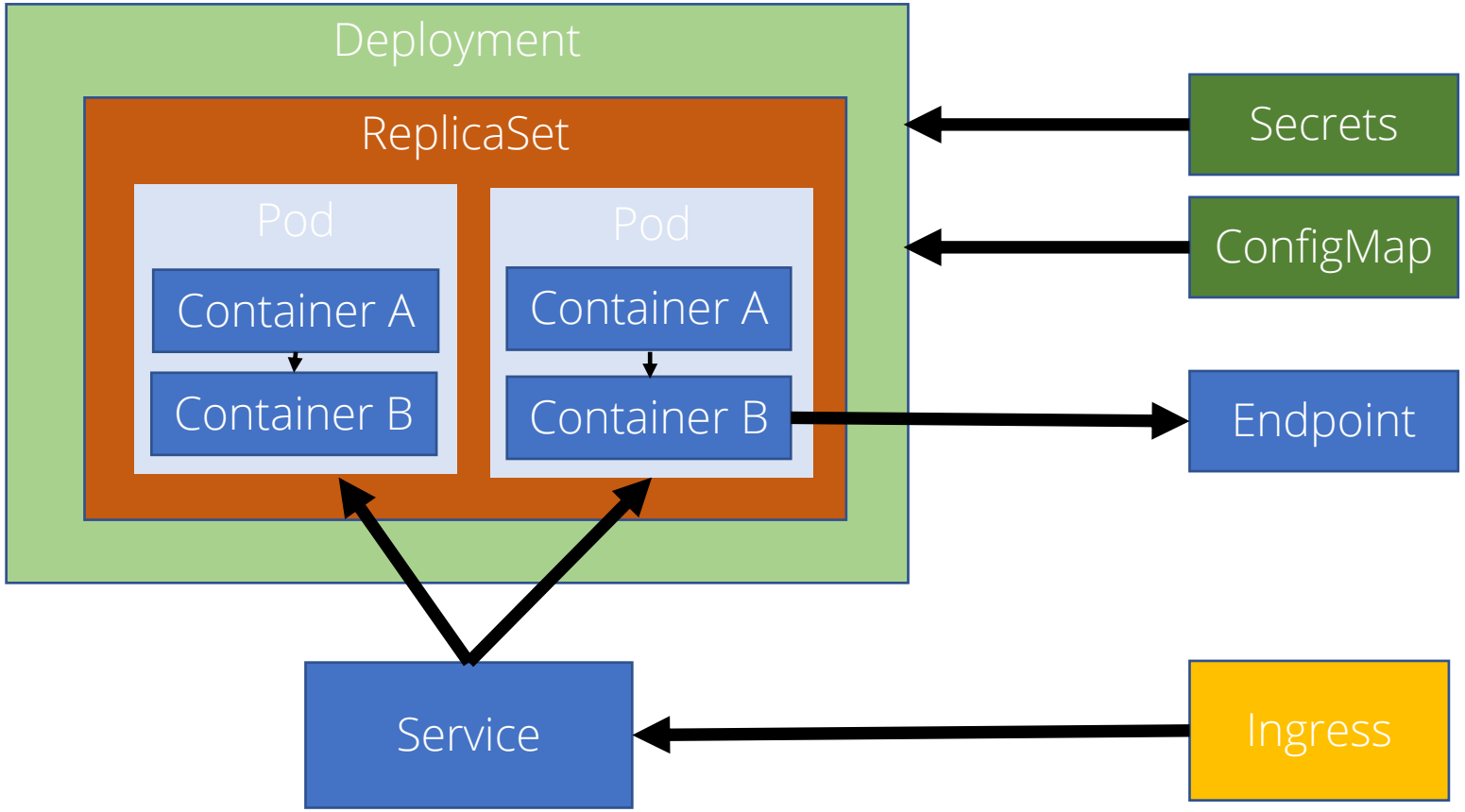
The many moving parts and new architecture are two hurdles to overcome when learning Kubernetes



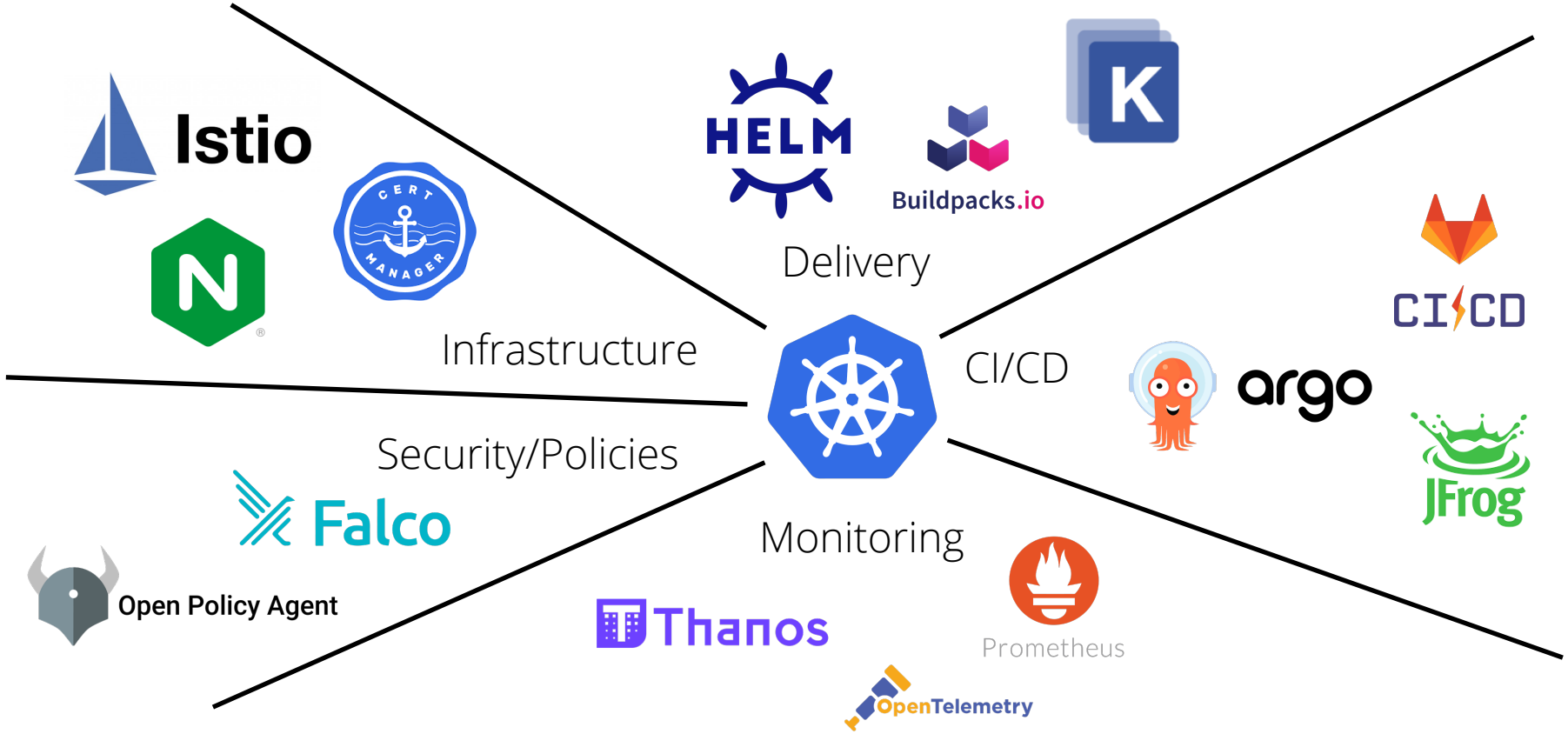
Incorrect configurations could also result in downtime, and this can be troublesome in a continuous integration and deployment environment

Developers are also unaware of the many ways code can introduce vulnerabilities

Example: A Kubernetes Deployment



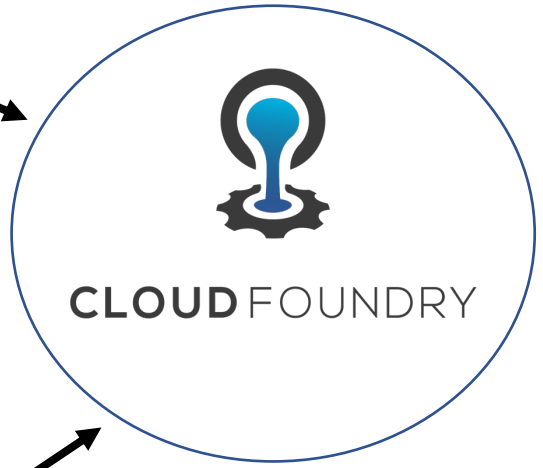
Ecosystem around Kubernetes



CF push

```
.
|-- bin
|-- build.gradle
|-- gradle
|-- gradle.properties
|-- gradlew
|-- gradlew.bat
|-- manifest.yml
|-- settings.gradle
`-- src
```

```
cf push -f manifest.yml
```

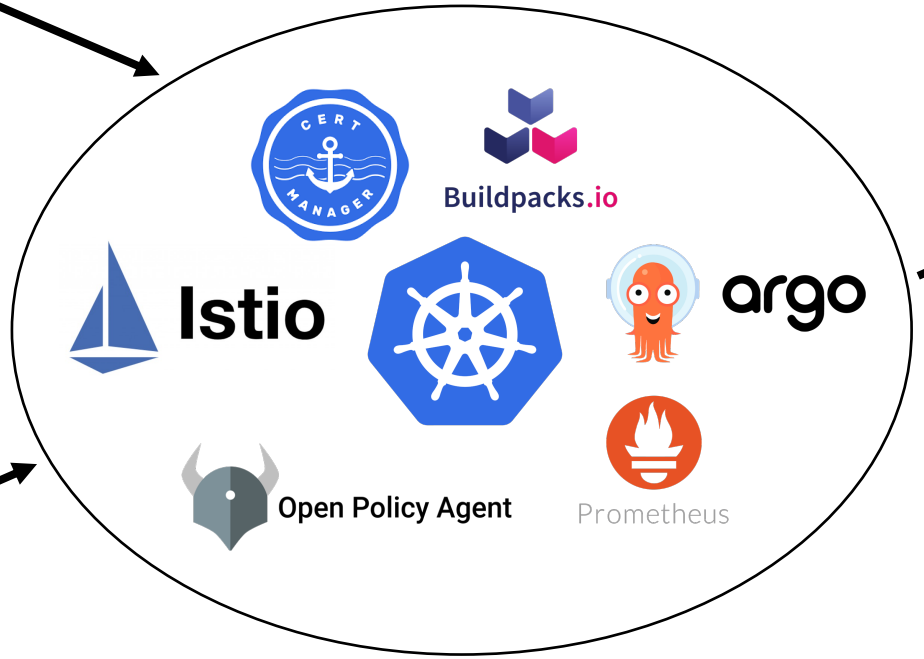


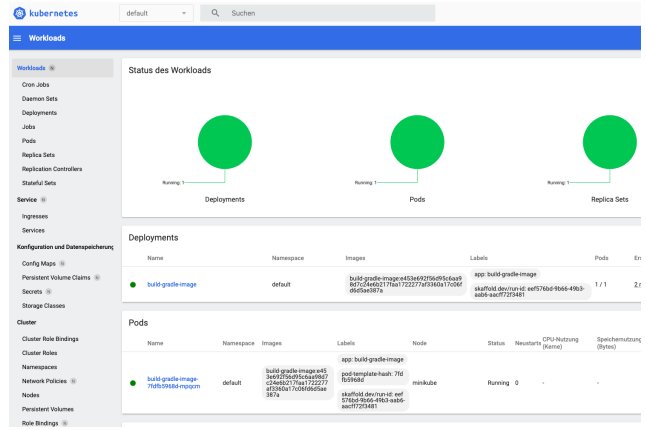

```
Creating app spring-music...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 607.75 KiB / 607.75 KiB
[==============================================================
==============================================================
==========] 100.00% 2s

Waiting for API to complete processing files...

Staging app and tracing logs...
   Downloading mgb_monitoring_buildpack...
   Downloading ruby_buildpack...
   Downloading staticfile_buildpack...
   Downloading java_buildpack...
   Downloading mgb_python_buildpack_latest...
   Downloaded java_buildpack (234.6K)
   Downloading dotnet_core_buildpack...
   Downloaded ruby_buildpack (5.2M)
   Downloading nodejs_buildpack...
   Downloaded staticfile_buildpack (5.7M)
   Downloading go_buildpack...
   Downloaded dotnet_core_buildpack (5.5M)
   Downloading binary_buildpack...
   Downloaded nodejs_buildpack (5.3M)
   Downloading mgb_php_oracle_buildpack_latest...
   Downloaded go_buildpack (5.2M)
   Downloading python_buildpack...
   Downloaded binary_buildpack (9.3M)
   Downloading mgb_ruby_buildpack...
   Downloaded python_buildpack (5.2M)
   Downloading mgb_go_buildpack_latest...
   Downloaded mgb_go_buildpack_latest
   Downloading php_buildpack...
   Downloaded php_buildpack (7.2M)
   Downloading nginx_buildpack...
   Downloaded nginx_buildpack (8.3M)
   Downloading r_buildpack...
```

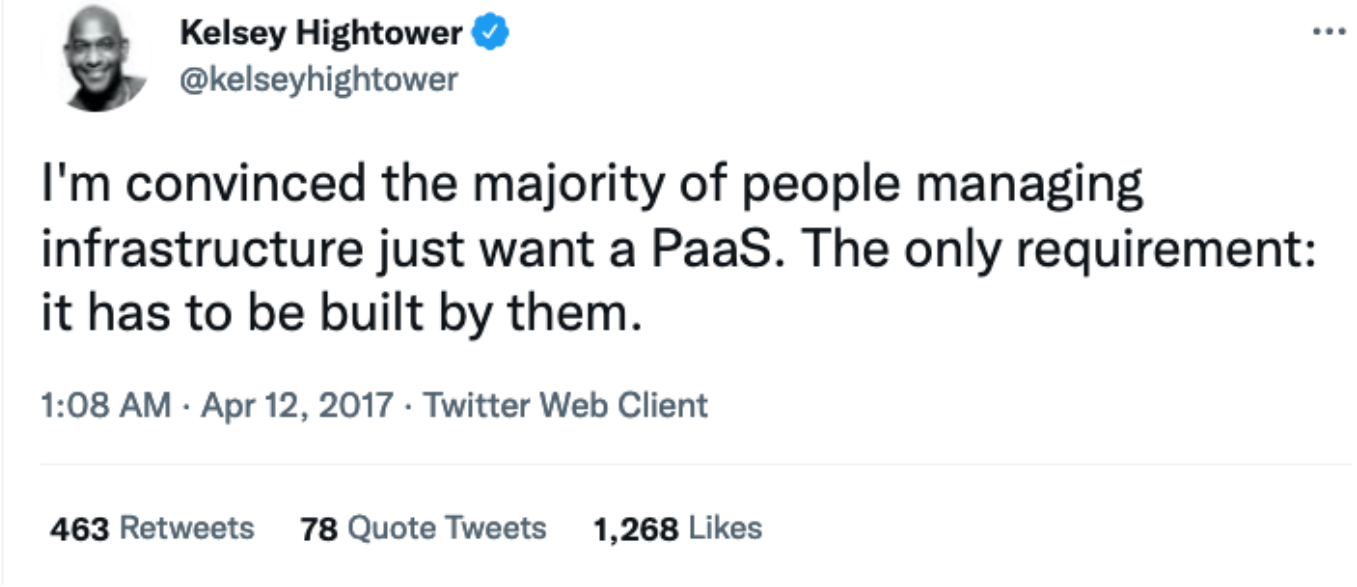
Kube push

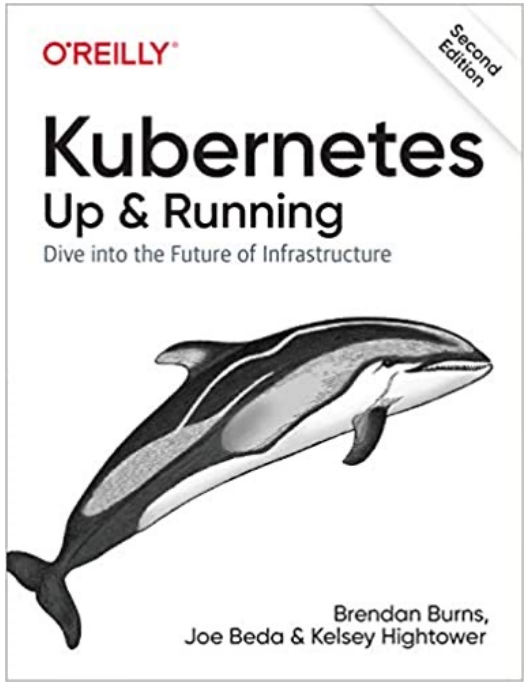
```
kubectl -n demo apply -f hello-world.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
  labels:
    app: hello-world
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-world
        image: hello-world
```



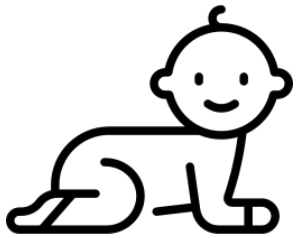






Source: https://twitter.com/kelseyhightower/status/851935087532945409?s=20&t=qmcl_X-XfUPxBvi_lD3Ndw

Maturity - Crawl / Walk / Run



Crawl-Stage

- Small team
- Junior level experience with Kubernetes
- Concentration on local development

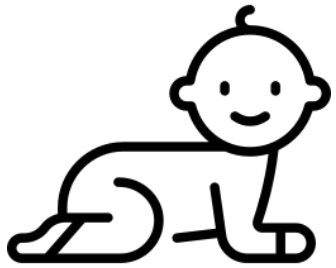
Walk-Stage

- Separation in dev and platform teams
- Workload becomes unwieldy
- Likely stitching collection of CNCF tools

Run-Stage

- Extending existing platform
- Inventing new building blocks

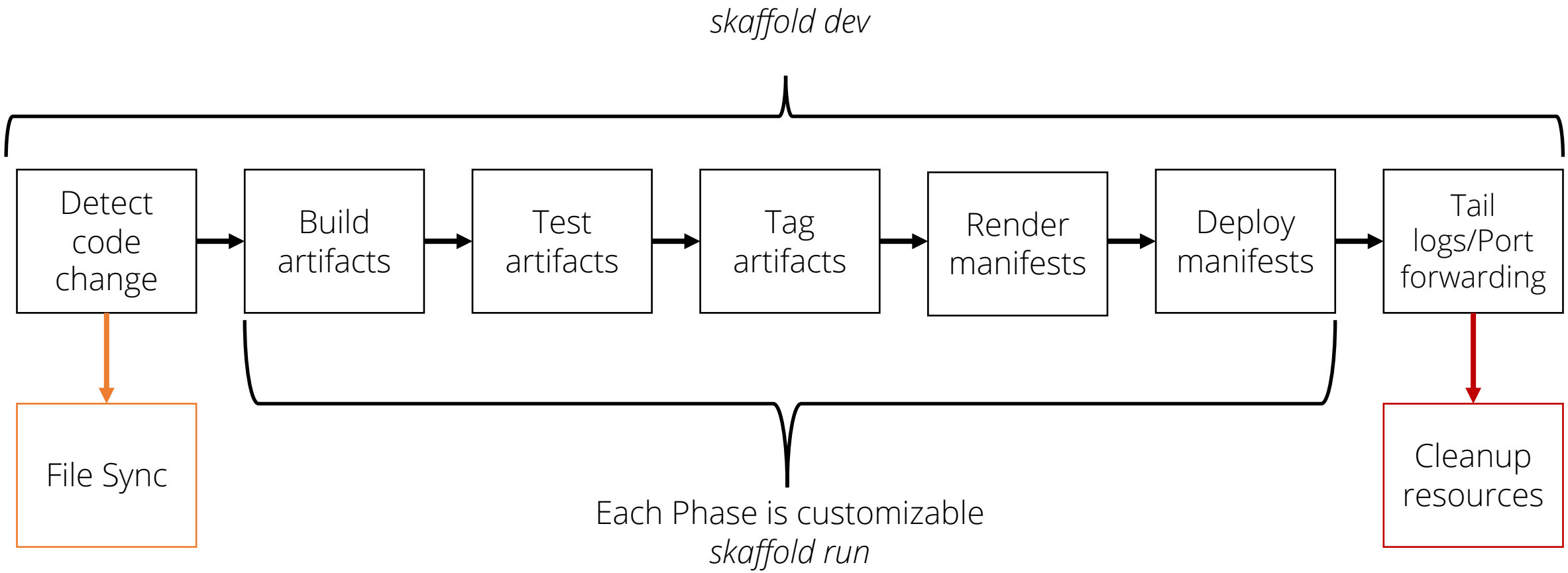
Crawl



Skaffold

- Handles the workflow for building, pushing, and deploying the application
- It automatically builds, creates and tags artifacts
- Provides building blocks for creating CI/CD pipelines
- The goal is to enable developers to focus on application development, not on acquisition of in-depth knowledge of Kubernetes
- Similar tools are DevSpace and Tilt

Skaffold Workflow



SKAFFOLD

Demo

Walk

Epinio

- Epinio is a Kubernetes-native platform as a service (PaaS)
- Aims to make it easy for every organization to have a platform to satisfy developers
- Uses cert-manager in order to create TLS certificates for the various Ingresses
- Any available paketo buildpack or pre-built images can be used
- Maintained by SUSE folks

EPINIO

Demo

Ketch

- Ketch allows developers to focus on the application code instead of the underlying infrastructure
- Frameworks: Allows Platform Engineers to isolate workloads across different namespaces
- Integration to Crossplane, Terraform and Pulumi

Demo

..and what about Cloud Foundry?

- KubeCF uses Fissile to turn cf-deployment BOSH releases into container images.
- Contributed from SUSE and IBM. Is not maintained anymore

- CF-For-K8s replaces almost all of Cloud Foundry with Einiri as backend. Example: Routing based on Istio, logging with Fluentd. Still on incubation
- Cloud Foundry Korifi: New architecture, still on incubation

Run

Considerations

- namespace-as-a-service or cluster-as-a-service
- Pave the paint points
- Documentation/Community are essential
- Maintain a product mindset
  - Make data-driven decisions
  - Continuous evolution

Conclusion

- The developer experience is about minimising the friction from idea to delivery observable business value
- The construction of the ‘platform’ impacts the developer experience greatly
- Building a better digital work space will help teams to be happier and more productive. This increases the quality of releases

Thank you!

kubectl apply –f questions.yaml