

# Kubernetes and The GitOps Face-Off

branch



**KubeCon**



**CloudNativeCon**

North America 2018

Javeria Khan & Ricardo Aravena

# Ricardo Aravena (rico)

Sr .Data Ops Engineer

 @raravena80, <raravena@branch.io>

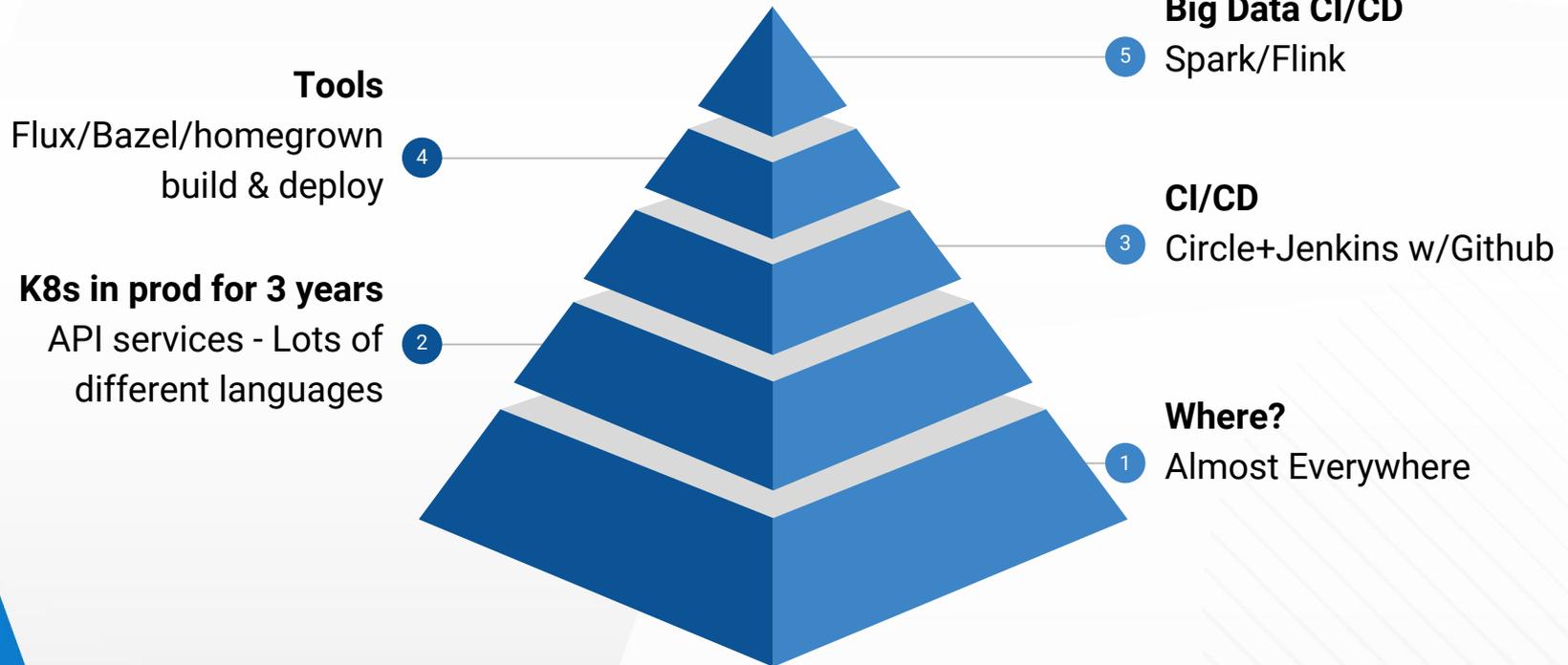
# Javeria Khan

Sr. Systems Engineer

 @javeriak\_, <jkhan@branch.io>



# K8s and GitOps @ Branch



# Agenda

## Background

- Our Scale & Usage
- GitOps History

## Tools

- Why use
- Scaffold
- Popular tools Pros & Cons
- Side by Side

## (Build) tools

- Why use
- Popular Tools
- Side by Side

## Choosing

- Production Ready
- Ease of use
- Community Support
- Stability

## Future

- Enhancements



# branch

---

**8 B** requests a day (+70% y/y)

**3B** user sessions per day

**100K** requests per second

**10** TB of data per day

**200+** microservices

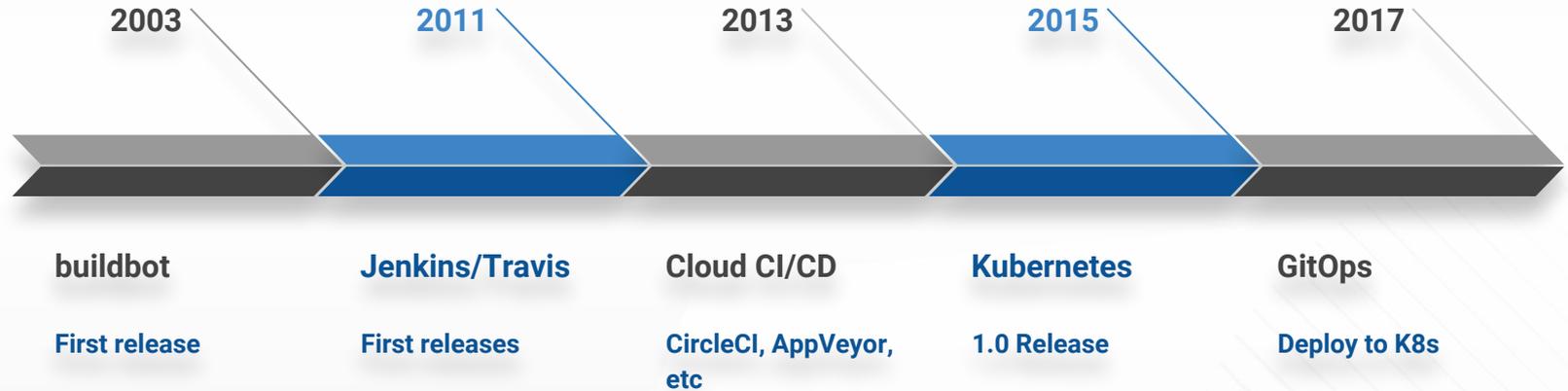
**~300** builds per day

**10** Kubernetes clusters

**10,000s** containers per cluster



# GitOps History



GitOps



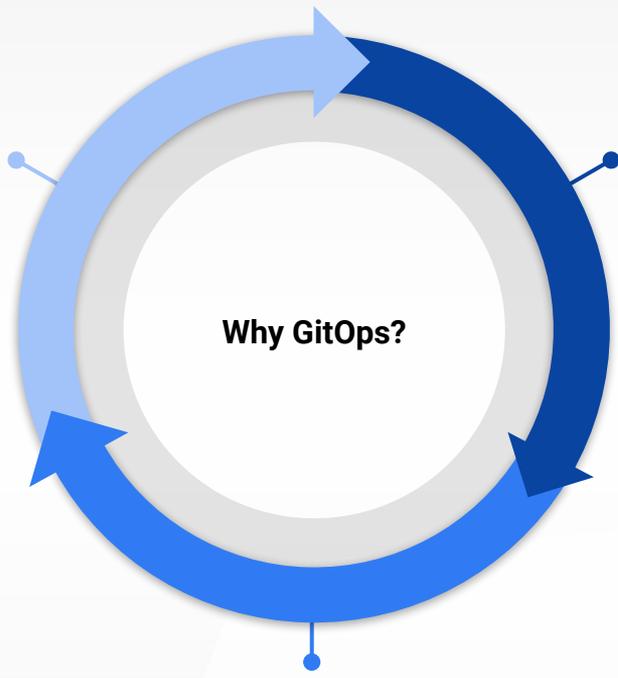
**Lots of Options!**



# GitOps Tools

---





**Why GitOps?**

Version Control  
**Both Infra & Code**

Use a tool  
**Compare current  
state to desired state**

Automate Deployments

**Gain reversibility,  
have an audit trail and  
transparency**



# Kubernetes @ Branch

In Production since **2016**

**10** kubernetes clusters

Run builds every **~5 min**

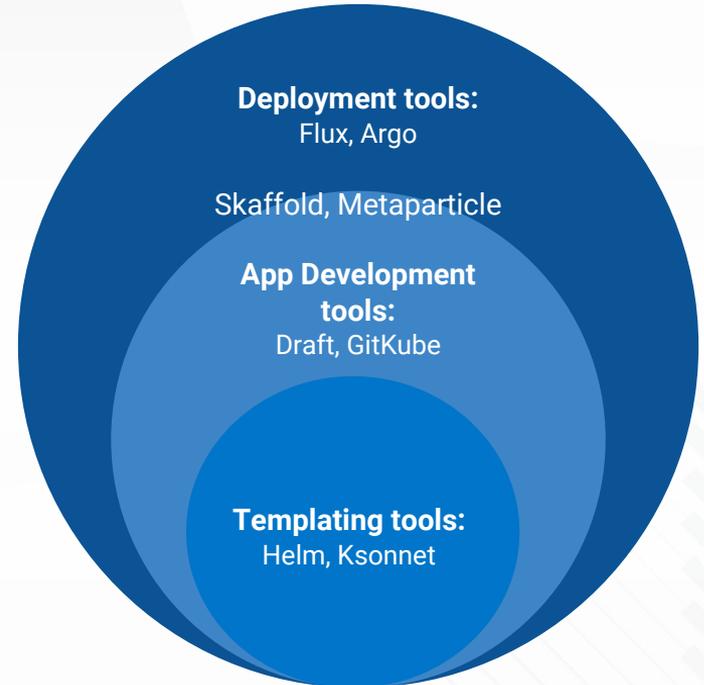
**10,000s** containers per cluster





# Desired Features

- ✓ Dependencies
  - Learning curve
- ✓ Handles source code
  - Iterative development
- ✓ Local development/test, remote cluster
- ✓ Logging
- ✓ Stability / Support (alpha, beta)
- ✓ Production use



“

# Ksonnet

*“YAMLs is for computers. Ksonnet is for people”*

<https://github.com/ksonnet/ksonnet>

# Ksonnet



Who?

- Heptio/Bitnami

What?

- Defining Apps

How?

- Jsonnet manifests

Dockerfile requirement

- No (uses existing images)

Docker daemon required

- No

Local and remote support

- Yes





## Pros

- **Modularity:** Dependency package management (github, filesystems, helm repos)
- **Support:** active community
- Supports deploying multiple versions to multiple clusters

## Cons

- **Dependencies:** knowledge of jsonnet
- **Stability:** early 0.13 release
- **Source Mgmt:** Does not handle source code updates
- No integrated image builder



“

Draft

*“Streamlined Kubernetes Development”*

<https://github.com/Azure/draft>

# Draft



Who?

- MS Open Source

What?

- App development

How?

- Draft packs/cli

Dockerfile requirement

- No (can use)

Local and Remote

- Yes

Docker daemon required

- Yes, local



# Draft Languages



Clojure



C#



Erlang



Go



Java



Python



Gradle



Javascript



PHP



Ruby



Rust



Swift





## Pros

- **Dependencies:** No Dockerfile requirement, No K8s manifests needed
- **Support:** active community
- Supports many languages
- Integrated docker image builder

## Cons

- **Dependencies:** Helm + Tiller, local docker daemon
- **Stability:** 0.16.x experimental release
- **Source Mgmt:** No automated updates to remote



“

# GitKube

*“Build & Deploy using git push”*

<https://github.com/hasura/gitkube>

# GitKube



Who

- Hasura

What?

- App Deployment

How?

- Git push

Dockerfile requirement

- Yes

Docker daemon required

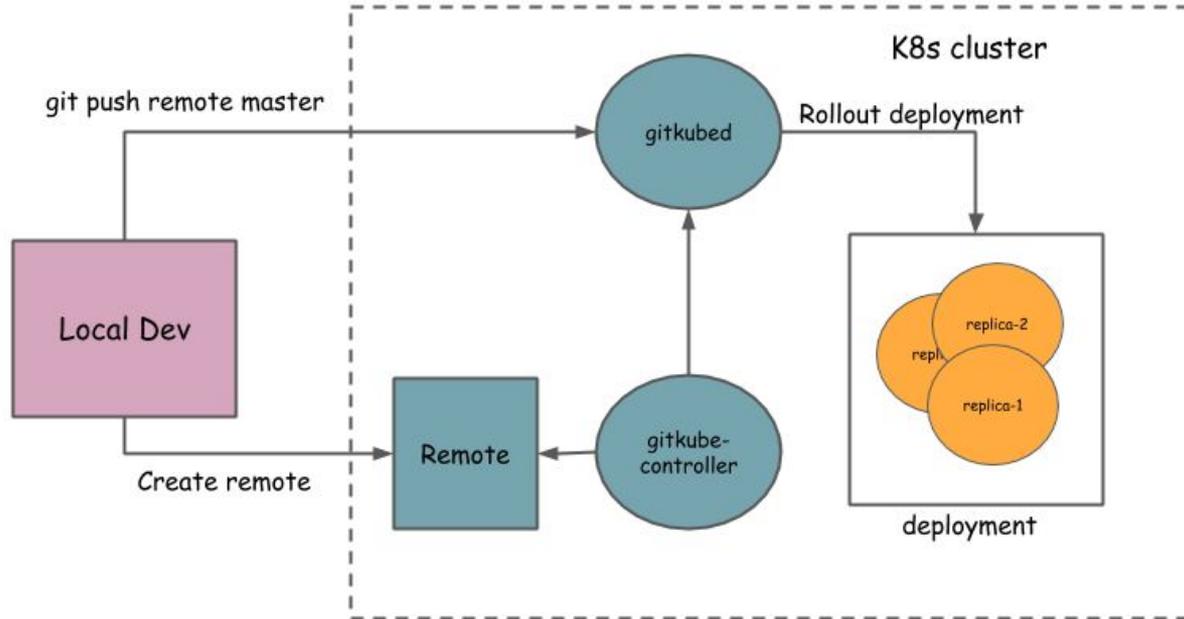
- Yes (in cluster)

Local and Remote

- Yes.



# GitKube



# GitKube



## Pros

- **Dependencies:** Uses existing common tools (git, kubectl)
- Easy setup
- Supports RBAC
- Supports any language
- **Source Mgmt:** Handles source code
- No cli, runs on cluster

## Cons

- **Dependencies:** Needs remote Docker, Dockerfile, k8s manifests in repo, CRDs
- **Stability:** Early release 0.2.1, no active community



“

Flux

*“Achieve Continuous Delivery and Integration”*

<https://github.com/weaveworks/flux>

# Flux



Who

- Weaveworks

What?

- App Deployment

How?

- git push
- fluxctl cmd

Dockerfile requirement

- No

Local and remote

- git for local
- fluxctl for remote

Docker Daemon required

- No





## Pros

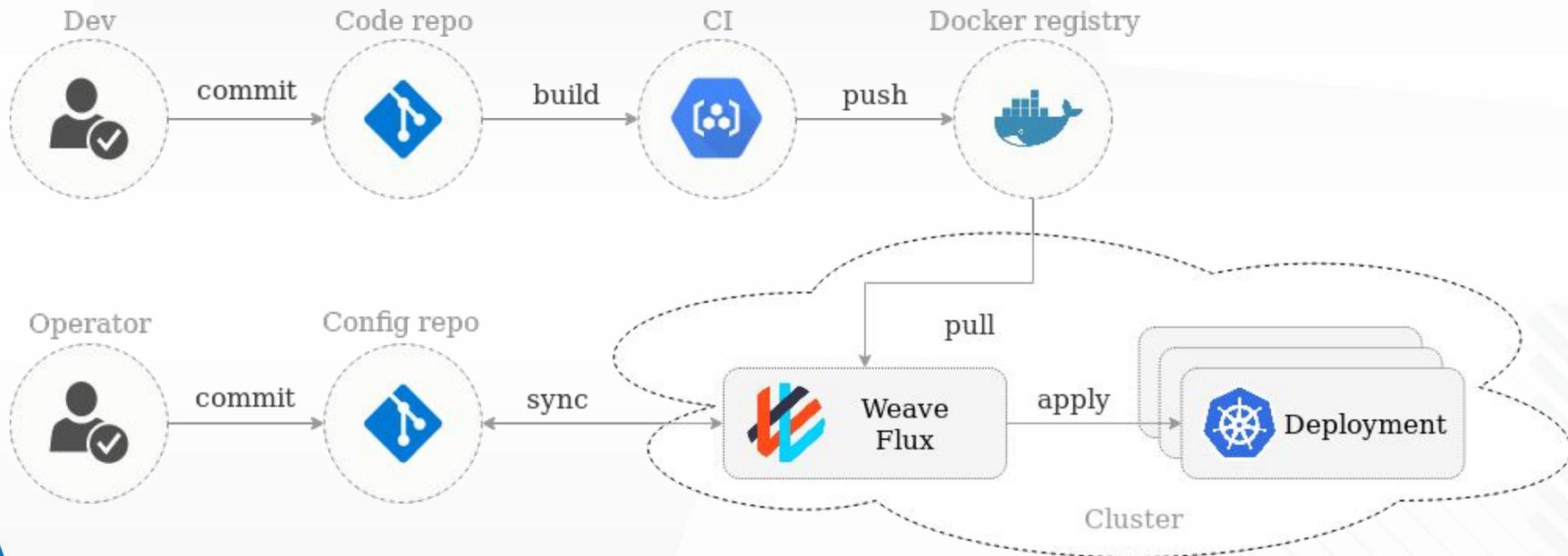
- **Dependencies:** Easy to deploy controller and CRD, supports Helm charts
- **Source Mgmt:** Automatic sync with docker registry and repo
- **Stability:** Mature 1.8.x release
- **Support:** active community

## Cons

- No local git management
- No integrated docker image builder
- Does it scale?
- Rollbacks possible but hard
- Canaries, blue-green hard



# Flux



“

# Skaffold

*“Easy and Repeatable Kubernetes Development”*

<https://github.com/GoogleContainerTools/skaffold>



# Skaffold

Who?	<ul style="list-style-type: none"><li>• Google Cloud</li></ul>
What?	<ul style="list-style-type: none"><li>• App Development &amp; Deployment</li></ul>
How?	<ul style="list-style-type: none"><li>• Watches git repo</li><li>• Git push</li></ul>
Dockerfile requirement	<ul style="list-style-type: none"><li>• Yes</li></ul>
Local and remote	<ul style="list-style-type: none"><li>• Yes</li></ul>
Docker daemon required	<ul style="list-style-type: none"><li>• Yes. Local and remote</li></ul>





# Skaffold

## Pros

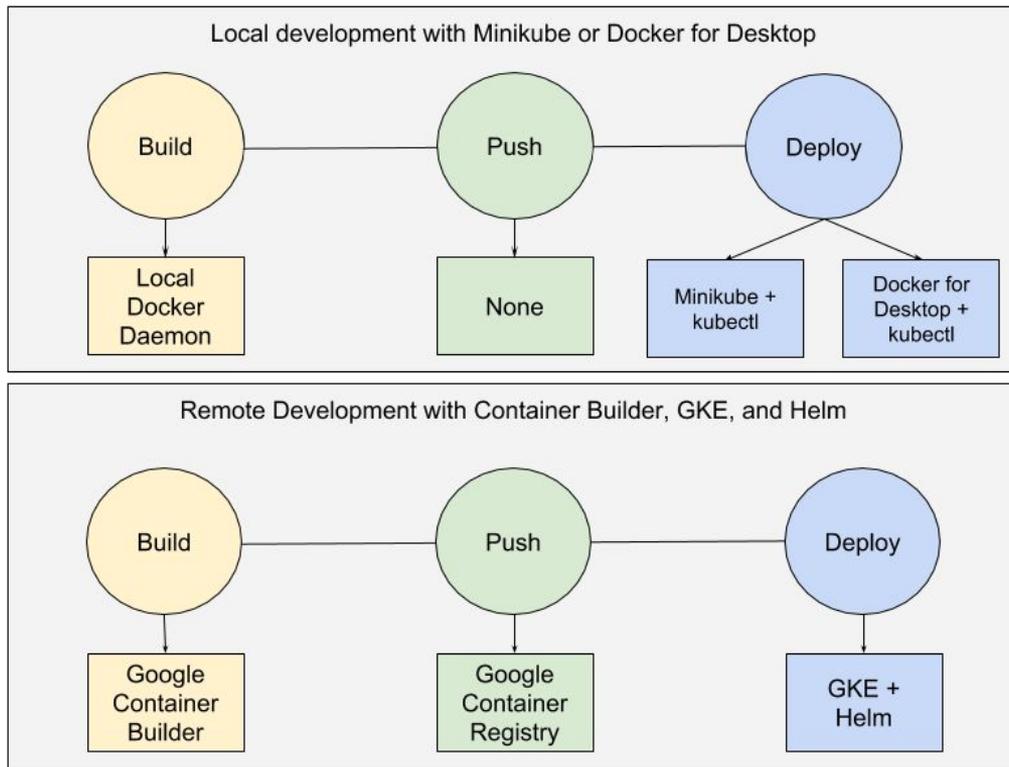
- Supports local and remote
- Has controller
- Supports any language/tool
- **Support:** Active community
- **Source Mgmt:** Automated updates from source code
- Integrated docker image builder

## Cons

- **Dependencies:** Needs local Docker, Dockerfile, Helm, K8s manifests in repo
- **Stability:** Early release 0.16.x



# Skaffold



“

Argo

*“Open source Kubernetes native workflows, events, CI & CD”*

<https://github.com/argoproj/argo>

# Argo



Who?

- Applatix

What?

- CI/CD

How?

- K8s Controller and CRDs

Dockerfile requirement

- No

Local and Remote

- No local docker build

Docker daemon required

- No



# Argo



## Pros

- Declarative YAML for Pipelines
- Nice UI
- Kubernetes integrated
- **Stability:** Mature 2.x release
- **Support:** Active development and community

## Cons

- No integrated docker image builder
- Complicated setup
- More of a workflow tool
- Cli is a wrapper for kubectl



“

# Metaparticle

*“Cloud Native standard library for Containers & Kubernetes”*

<https://github.com/metaparticle-io>

# Metaparticle

Who?

- Brendan Burns

What?

- Templating + Deployments

How?

- Using actual code

Dockerfile requirement

- No

Local and Remote

- Yes

Docker daemon required

- No



# Metaparticle

## Pros

- **Dependencies:** No Dockerfile, YAML or config files
- Code based deployment
- Infra as real code
- More language support coming
  - Go
  - Rust
  - Ruby

## Cons

- **Dependencies:** Needs local Docker
- Limited language support
- **Stability:** Very alpha, no community
- Idiomatic



# GitOps Tools Side by Side

Tool	Dockerfile requirement	Docker Daemon req	Function	Method	Local / Remote	Helm integration
Draft	No	Yes, in cluster	Deploy to K8s	Draft packs	Yes	Yes
Flux	Yes	No	Full lifecycle	git push fluxctl	Yes	Yes
GitKube	Yes	Yes, in cluster	Deploy to K8s	git push	Yes	
Skaffold	Yes	Yes, local & remote	Deploy to K8s	K8s YAML/JSON	Yes	
Argo	No, but can use	No	CI/CD	K8s YAML/JSON	Remote only	
Ksonnet	No, but can use	No	Deploy to K8s	Jsonnet	Yes	Yes
Metaparticle	No, but can use	Yes, local	Deploy to K8s	Code libs	Yes	

# Container Build tools

---

Kaniko

Img

Orca-build

Umoci

Buildah

FTL

Bazel Docker

# Why use something other than docker build?

Faster builds (most of the time)

Run unprivileged

- More secure

No need for DinD

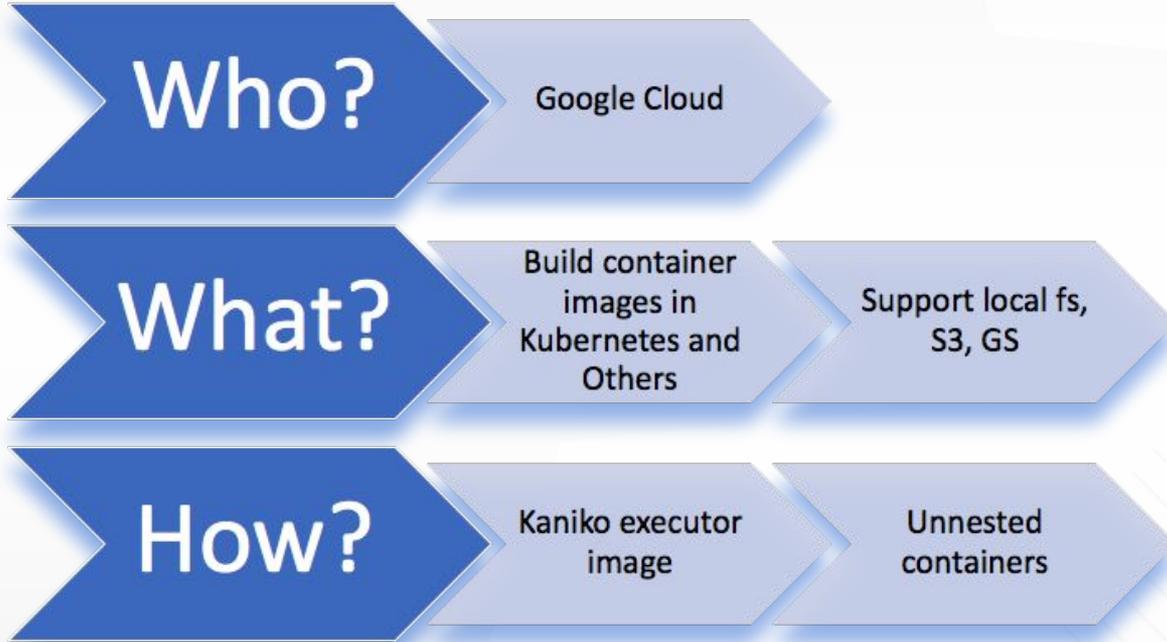
- Allows creation of Kubernetes based CI slaves that build images in pods

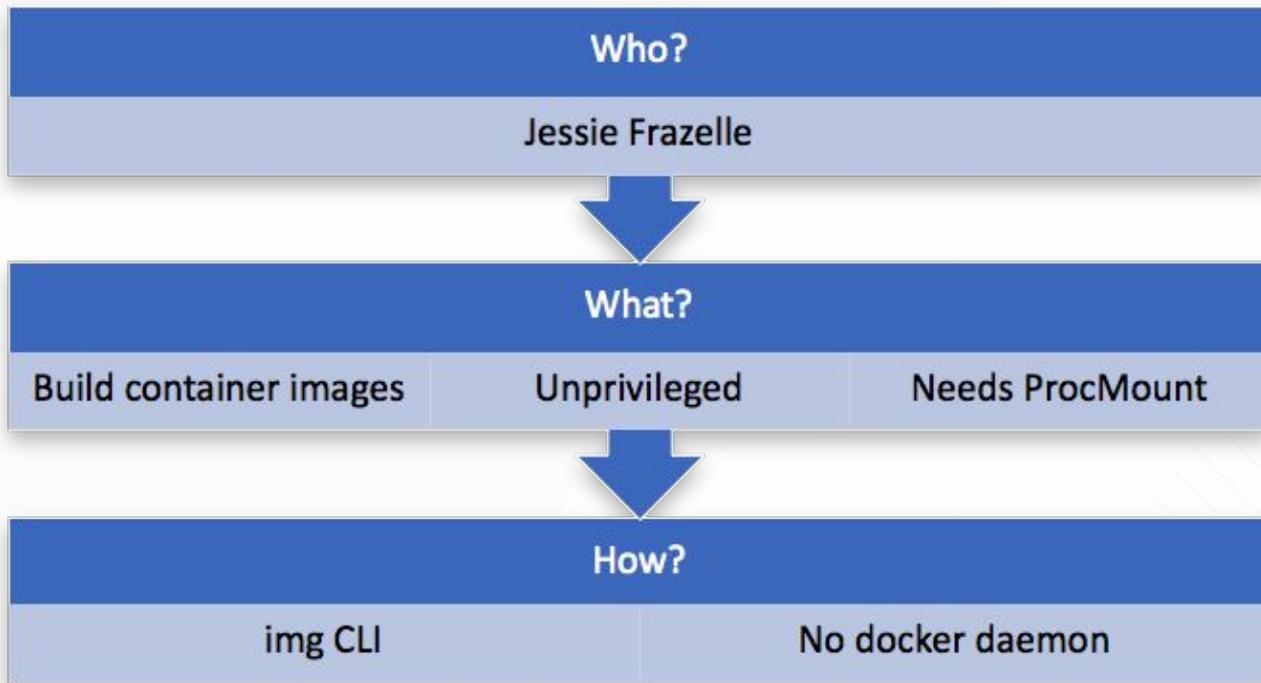
Reproducibility (Bazel)

- *“Running Make with an imperfectly written Makefile inside a Docker container can still yield unpredictable results.”*



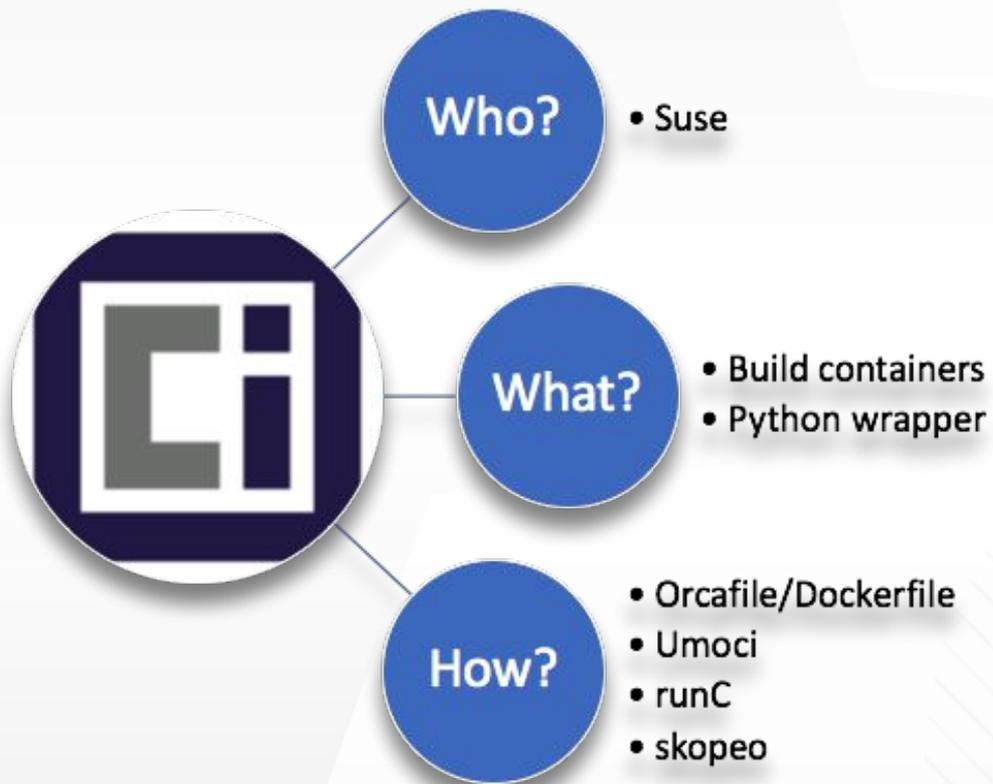
# Kaniko



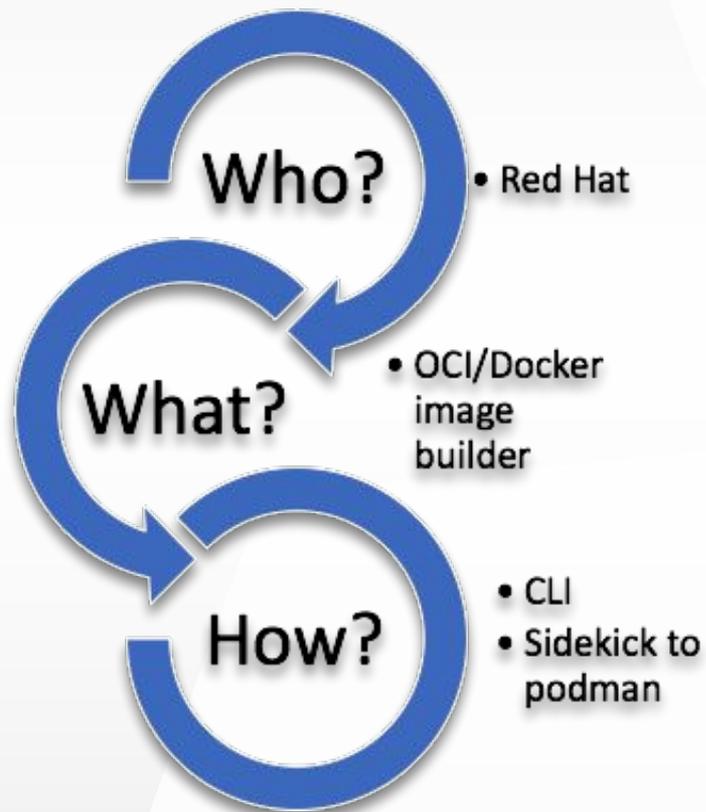




# Orca Build/Umoci

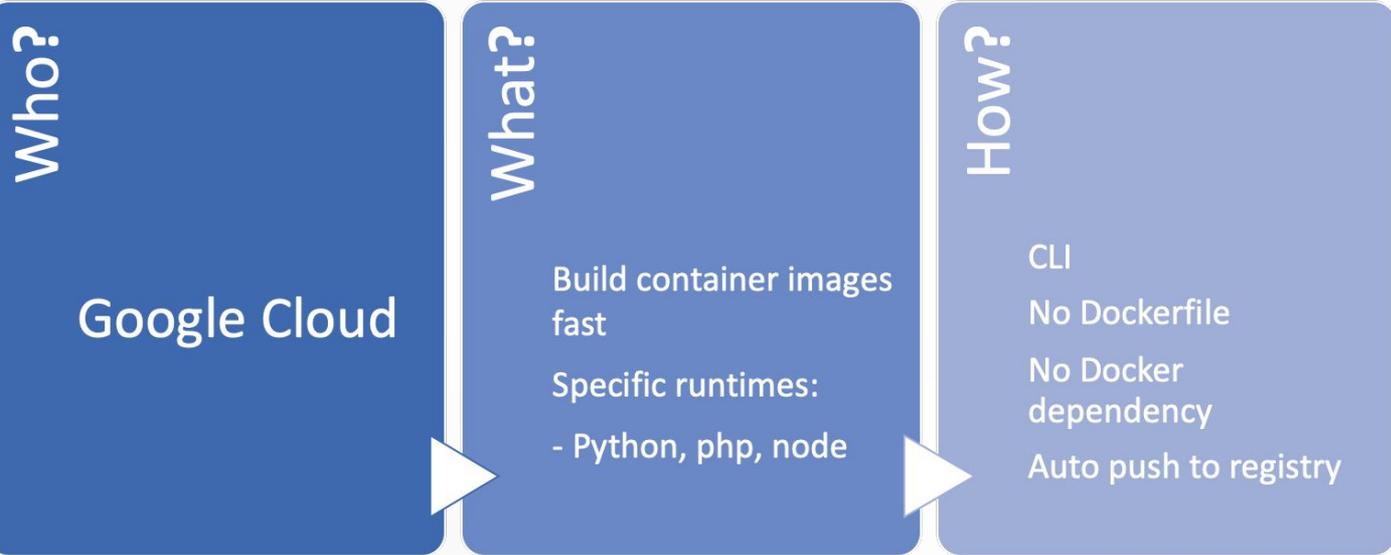


# Buildah



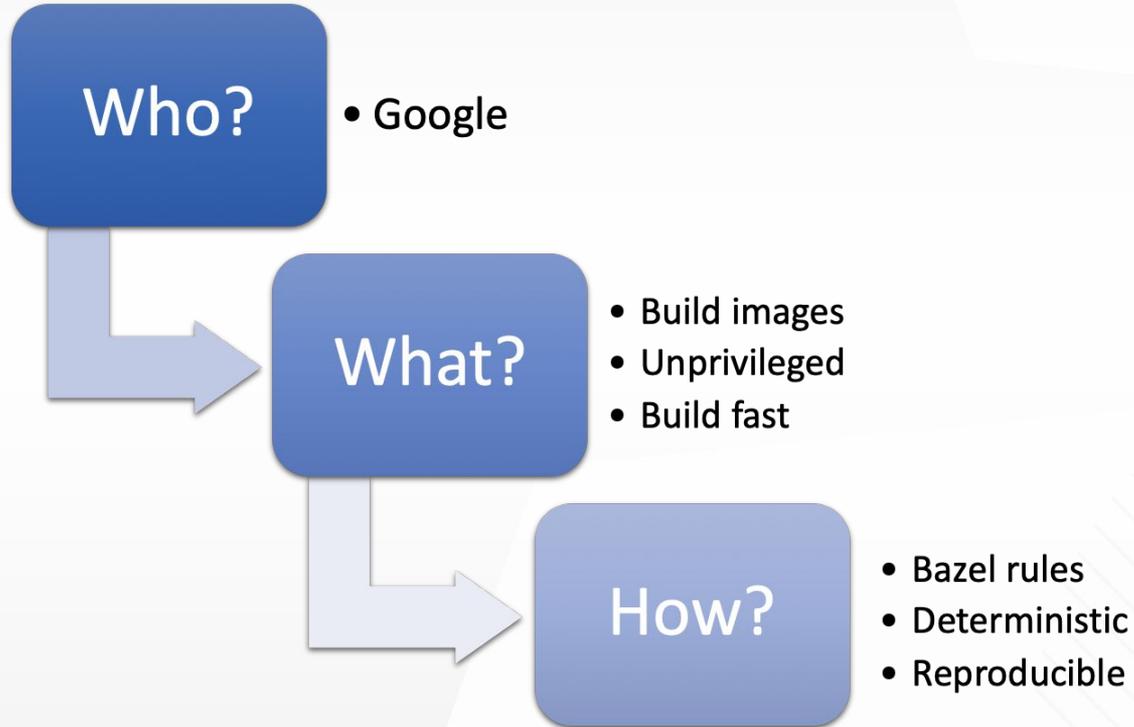


# FTL - Faster Than Light





# Bazel Docker/OCI



# Knative



<b>Who?</b>	<ul style="list-style-type: none"><li>• Google Cloud</li></ul>
<b>What?</b>	<ul style="list-style-type: none"><li>• Serverless in K8s</li><li>• Build templates</li></ul>
<b>Templates</b>	<ul style="list-style-type: none"><li>• Kaniko</li><li>• Bazel</li><li>• Buildah</li><li>• Others</li></ul>



# Image Build Tools Side by Side

Tool	Who?	What?	How?	Docker daemon	In K8s Cluster
Kaniko	Google Cloud	Build in K8s	Image builder, un-nested	No	Yes
Img	Jess Frazelle	Build Unprivileged	Dockerless, RawProc	No	Yes, nested
Umoci/Orca-build	SUSE	Just Build	Unprivileged, needs runC	No	Yes, nested.
Buildah	Red Hat	Just Build	Requires privilege escalation	No	Yes, w/Knative
FTL	Google Cloud	Just Build	Layers/Dockerless	No	Yes, with Kaniko
Bazel	Google	Just Build	Bazel definition	No	Yes, w/Kaniko-Knative
Knative	Google Cloud	Build templ. In K8s	Using templates	No	Yes, requires istio

# GitOps with Build Tools

GitOps Tool	Build Tools
Flux	✓
Ksonnet	✓
Metaparticle	✓
Argo	✓
Draft	✗
GitKube	✗
Skaffold	✗

# What to use in prod?



# What to use in prod?

Easy

## Gitops

- Draft
- Skaffold
- Flux



## Image Builds

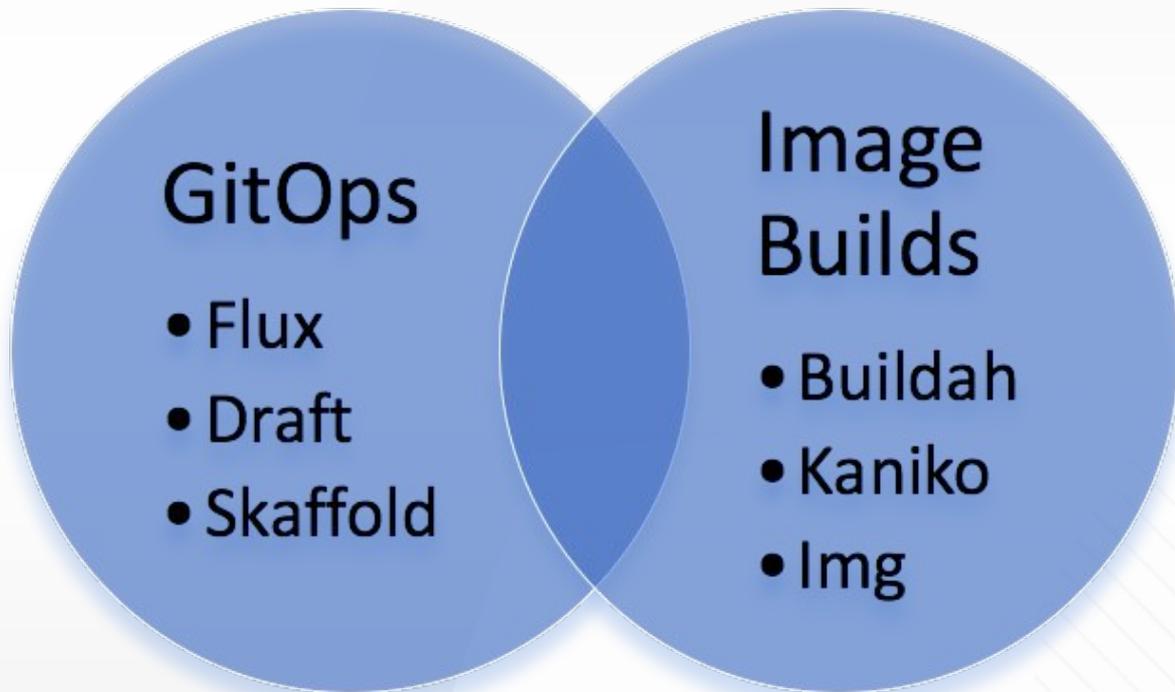
- Img
- FTL



Easy

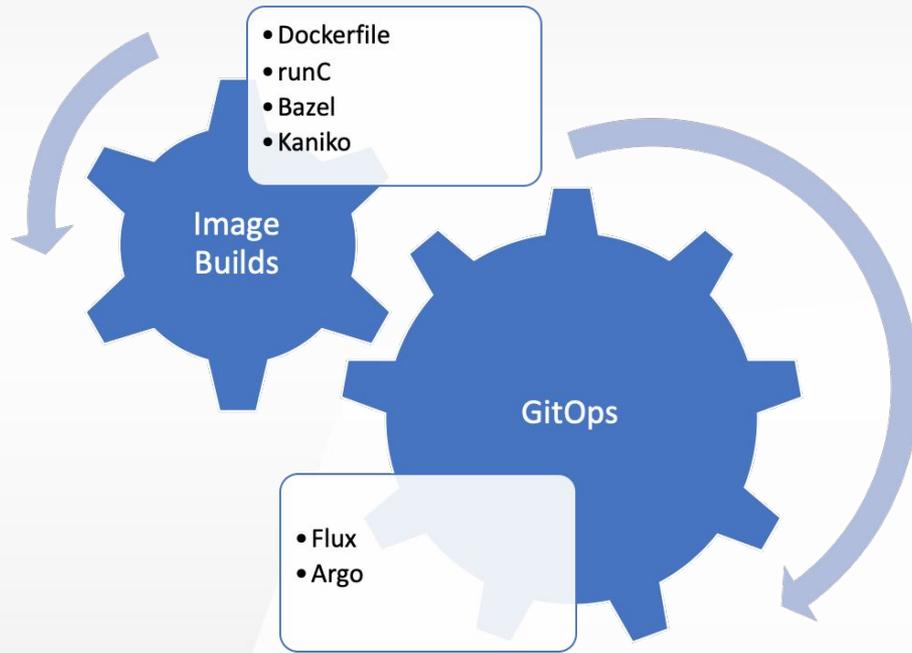


# What to use in prod?



# What to use in prod?

Production Ready



Stable



# Future

## GitOps

- 1.0 releases
- New tools
- CRD integration and development

## Build Tools

- 1.0 releases
- More tools to build with Kubernetes
- Direct integration with K8s

## Helm

- Helm v3
- Lua plugins/hooks
- Event driven architecture



# Resources

Scaffold	<a href="https://github.com/GoogleContainerTools/skaffold">https://github.com/GoogleContainerTools/skaffold</a>
Draft	<a href="https://github.com/Azure/draft">https://github.com/Azure/draft</a>
Flux	<a href="https://github.com/weaveworks/flux">https://github.com/weaveworks/flux</a>
GitKube	<a href="https://github.com/hasura/gitkube">https://github.com/hasura/gitkube</a>
Argo	<a href="https://github.com/argoproj/argo">https://github.com/argoproj/argo</a>
Ksonnet	<a href="https://github.com/argoproj/argo">https://github.com/argoproj/argo</a>
Kaniko	<a href="https://github.com/GoogleContainerTools/kaniko">https://github.com/GoogleContainerTools/kaniko</a>
Img	<a href="https://github.com/genuinetools/img">https://github.com/genuinetools/img</a>
Orca build/Umoci	<a href="https://github.com/openSUSE/umoci">https://github.com/openSUSE/umoci</a>
Buildah	<a href="https://github.com/projectatomic/buildah">https://github.com/projectatomic/buildah</a>
FTL	<a href="https://github.com/GoogleCloudPlatform/runtimes-common/tree/master/ftl">https://github.com/GoogleCloudPlatform/runtimes-common/tree/master/ftl</a>
Bazel Docker/OCI	<a href="https://github.com/bazelbuild/rules_docker">https://github.com/bazelbuild/rules_docker</a>



# Thanks!

Branch Engineering is hiring!

<https://branch.io/careers/>

branch 