



# Cloud Native Development



RAMIRO BERRELLEZA | @RBERRELLEZA

# About @rberrelleza

---

- Co-founder of Okteto
- Former architect @ Atlassian, engineer @ Microsoft Azure
- <https://twitter.com/rberrelleza>



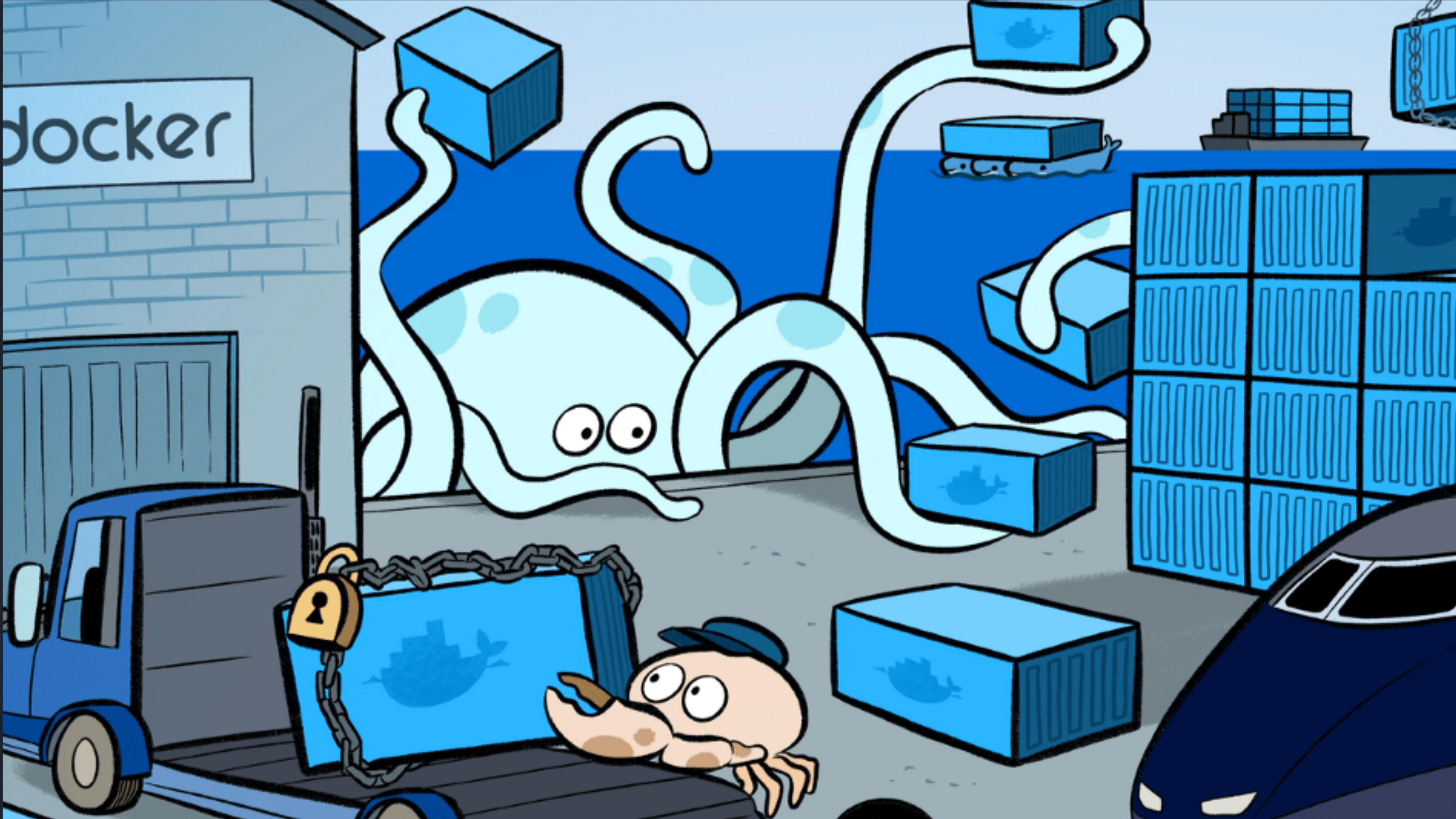
# Agenda

---

- Current state of application development
- Developing cloud native applications is hard
- Cloud Native Development makes it easier
- Demo



# Current state of application development



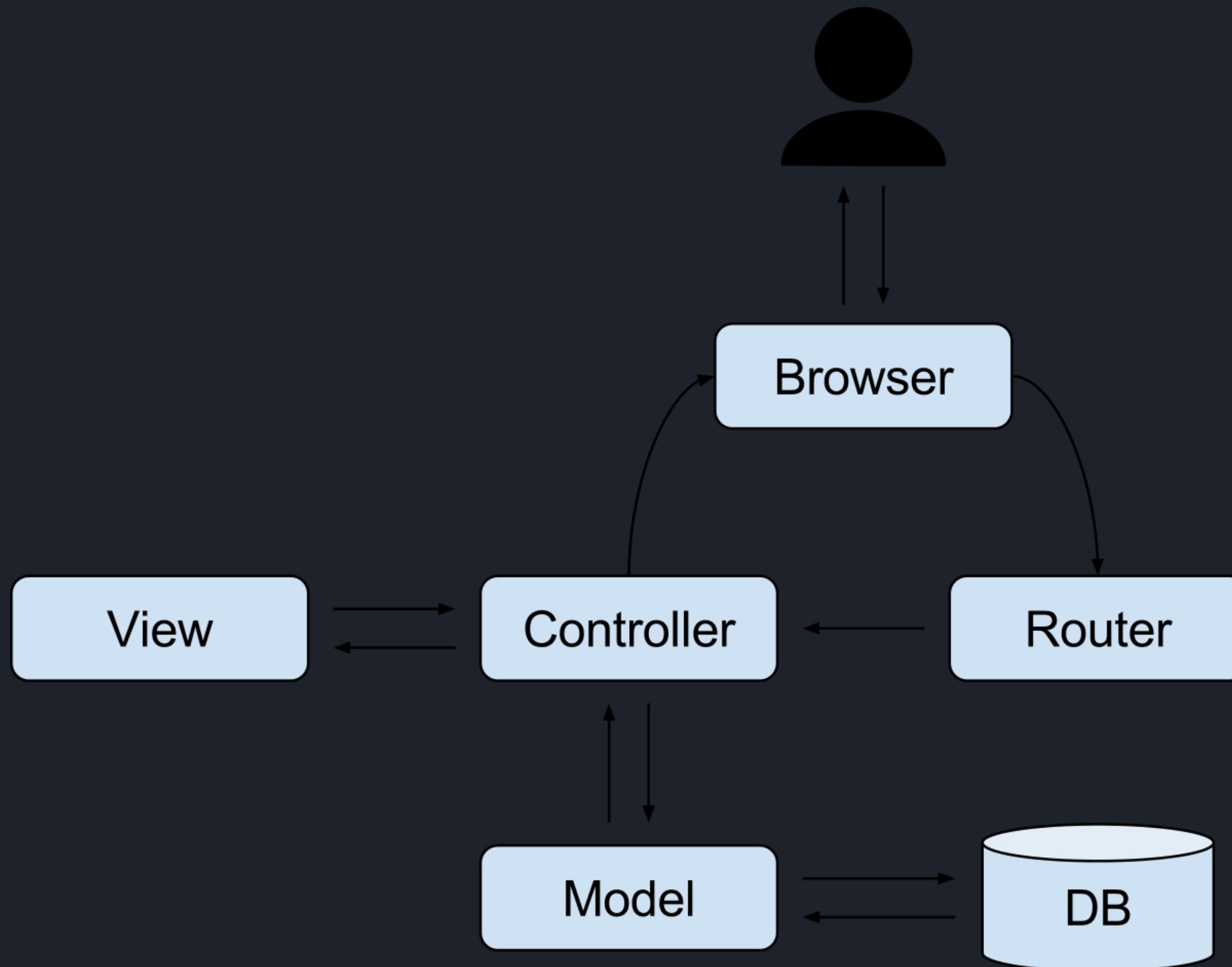
Supported by Google's  
internal systems  
environments  
supports multiple cloud and bare-metal  
0% **Open source**, written in Go  
ge applications, not machines

# Google Cloud



Docker and Kubernetes  
have made it easier to  
create and run applications  
at high scale ...

... at the expense of the  
developer experience

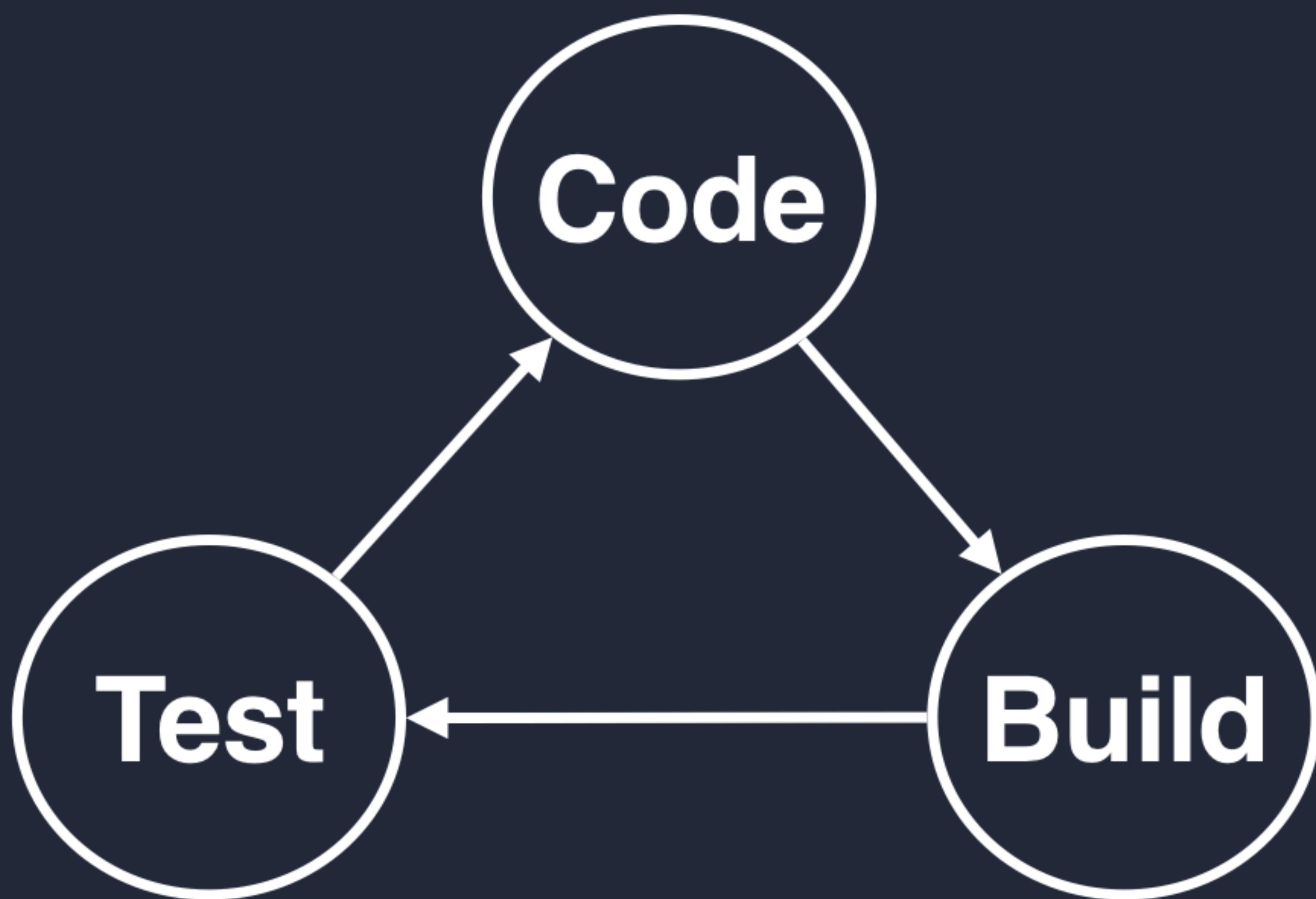


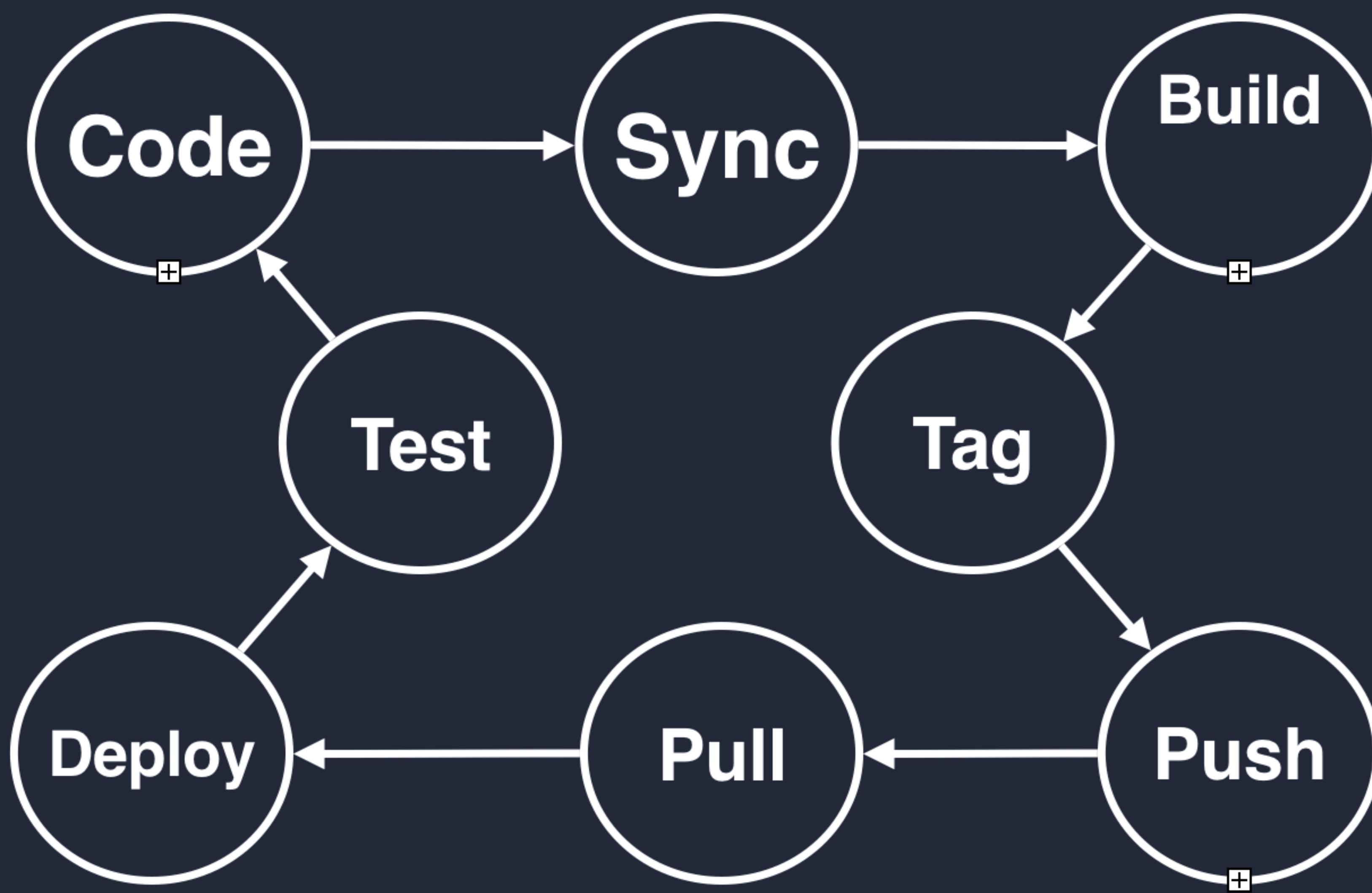




Developing cloud  
native applications is  
hard

An inner loop full of friction  
makes it harder to develop  
even the simpler features





The community is trying to  
solve this issue in different  
ways



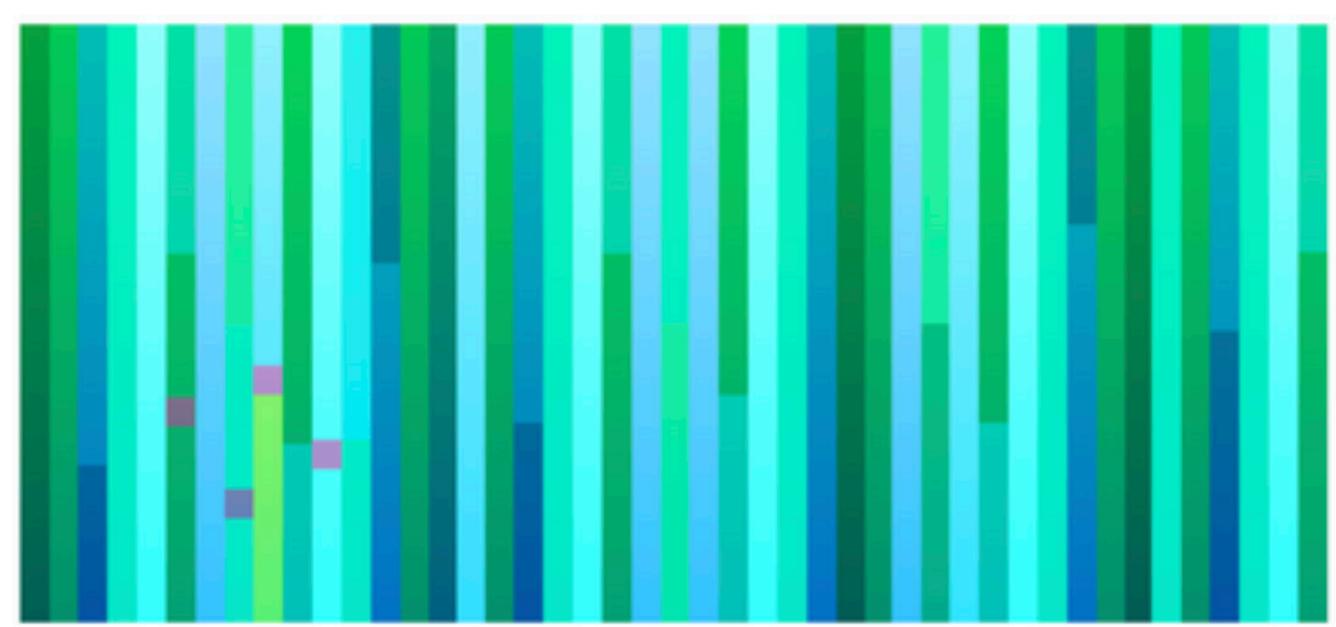
L

@ellenkorbes

Follow



Here are the slides for my talk, The State of  
Kubernetes Development Tooling, at  
[@ConDaysEU #CDS19 #ContainerDays!](#)



**The State of Kubernetes Development Tooling**  
A presentation created with Slides.  
[garden.slides.com](http://garden.slides.com)

2:32 AM - 25 Jun 2019

<https://twitter.com/ellenkorbes/status/1143451907492655105>

## But is still hard

---

- Development environment is not replicable.
- Manifest explosion (docker-compose, k8s manifests, Jenkins configurations, etc...)
- Environment explosion (dev, integration, stage, prod...)

## But is still hard

---

- Can't run the entire app locally due to size and dependency
- Shared integration environments that are typically always broken
- Too much responsibility on CI for end to end validation



Cloud Native  
Development makes  
it easier

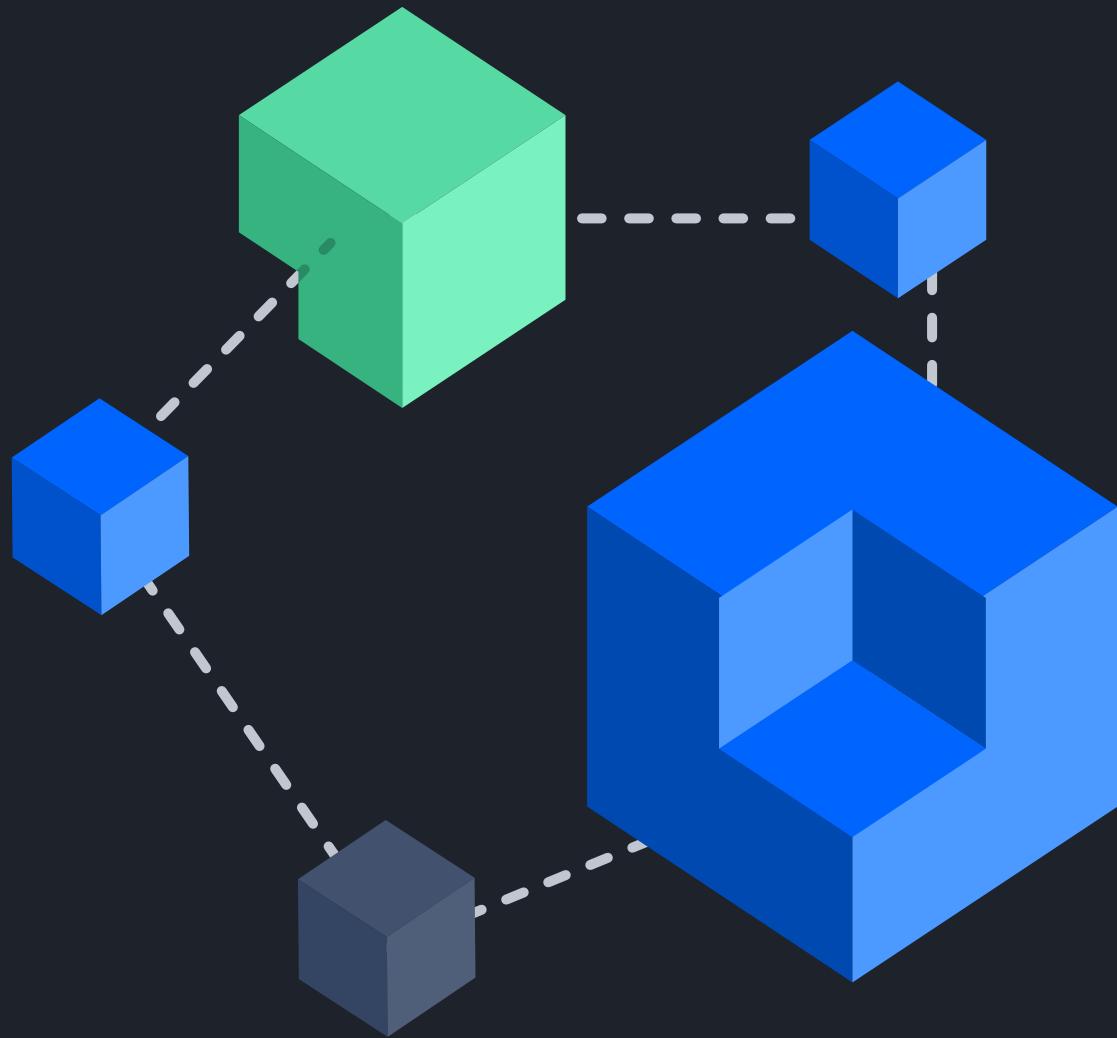


DEVELOP YOUR  
APPLICATIONS IN  
THE SAME  
ENVIRONMENT  
WHERE THEY ARE  
GOING TO RUN

After a couple of years of  
building cloud native  
applications, we ended up  
with this workflow



Use the same cluster  
configurations for dev  
and prod



Don't mock your  
dependencies, use  
them from the very  
beginning



Invest in an agile development workflow.

**Don't build/push/  
deploy on every  
change.**



## Reuse your artifacts.

The first phase of a multi-build container makes a great development environment.



Leverage Kubernetes  
on-demand compute  
and create ephemeral,  
on-demand dev  
environments

# We did found some challenges

---

- Developers need self-service access to dev clusters
- But sharing clusters means developers can get on each other's business
- We don't want everyone to be cluster-admin

# We did found some challenges

---

- Kubernetes SIG-Multitenancy is looking at some of these issues
- But we build tools to start solving some of the issue we had: <https://github.com/okteto/okteto>

ckteto

# Demo

Develop the Hipster Shop directly in the cluster

<https://github.com/GoogleCloudPlatform/microservices-demo>

## More resources

---

- Star it, fork it and collaborate: <https://github.com/okteto/okteto>
- Try it yourself at <https://cloud.okteto.com>
- Posts on cloud native development: <https://medium.com/okteto>
- Stay in touch! <https://twitter.com/rberrelleza>



Thank you!



RAMIRO BERRELLEZA | @RBERRELLEZA