okteto

Don't build and push, move your inner loop to your cluster!

RAMIRO BERRELLEZA | @RBERRELLEZA

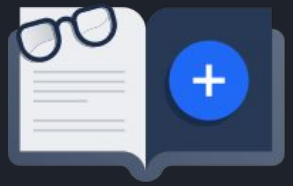
Hey everyone!

- Co-founder of Okteto.
- Former architect @ Atlassian, Software Engineer @ Azure.
- @rberrelleza in most places.

Application architecture has evolved a lot in the past few years...

June

Challenges



Dependency handling

Repeatable builds

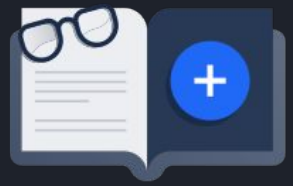
Automated deployments

Provisioning infrastructure

docker

by Google's
and internal systems
upports multiple cloud and bare-metal
nvironments
upports multiple container runtimes
00% Open source, written in Go
ge applications, not machines
Google Cloud

Challenges



Dependency handling

Repeatable builds

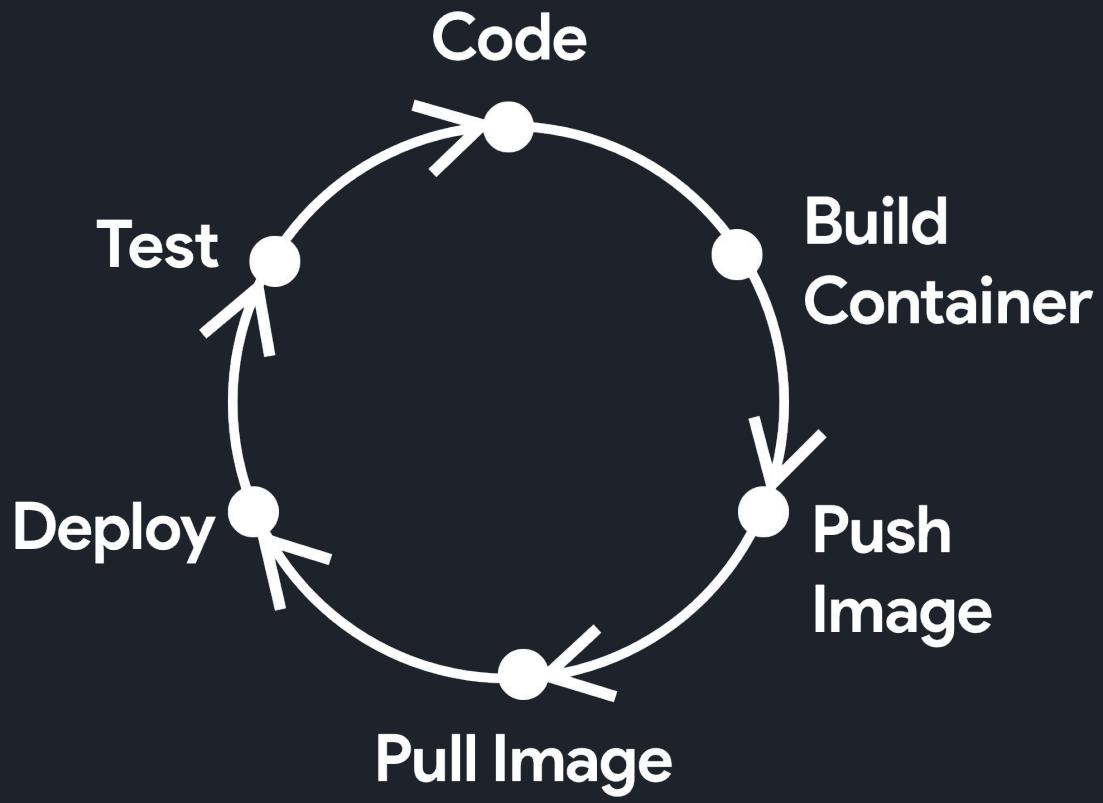
Automated deployments

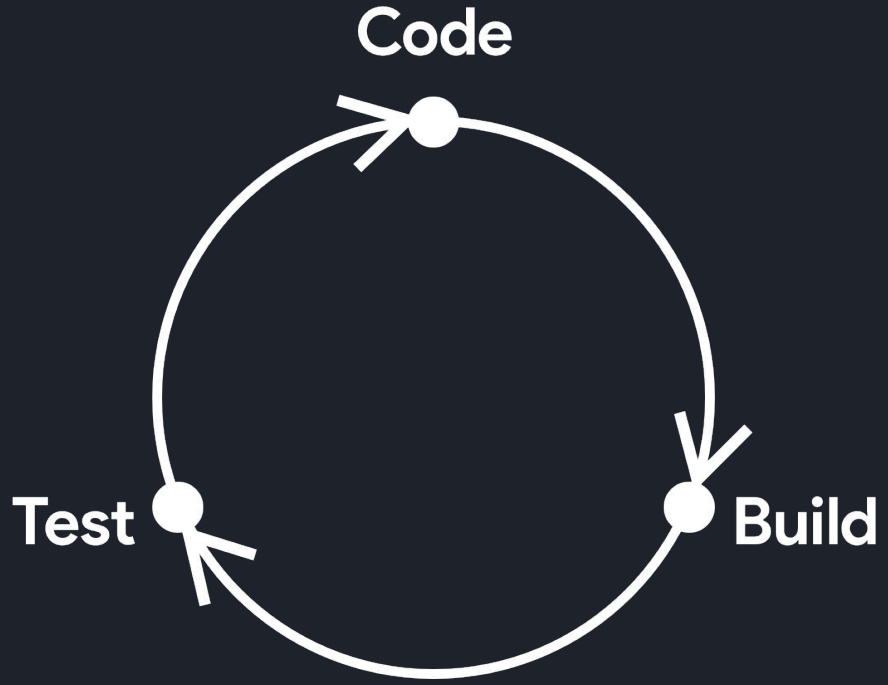
Provisioning infrastructure

We solved production*, but at the expense of the development experience.

Building Cloud Native Applications is hard.

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


Code
Build
Container
Push
Image
Pull Image
Deploy
Test


Code
Build
Test

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

Community

Automate the inner loop

- Skaffold
- Draft
- Garden
- Tilt

Community

Dev environments on demand

- Visual Studio Code Online
- Code Ready workspaces (Eclipse Che)
- Jenkins X

yay!

But we're still waiting on builds and deploys.

… and dev, stage and prod are completely different

We started Okteto to give us a better developer experience in Kubernetes

Okteto: A Tool for Cloud Native Developers

Overview

Kubernetes has made it very easy to deploy applications to the cloud at a higher scale than ever, but the development practices have not evolved at the same speed as application deployment patterns.

Today, most developers try to either run parts of the infrastructure locally, or just test these integrations directly in the cluster via CI jobs or the "docker build, docker push, kubectl apply" cycle. It works, but this workflow is painful and incredibly slow.

Okteto makes this cycle a lot faster by launching your development environment directly in your Kubernetes cluster.

Features

Development environments on demand

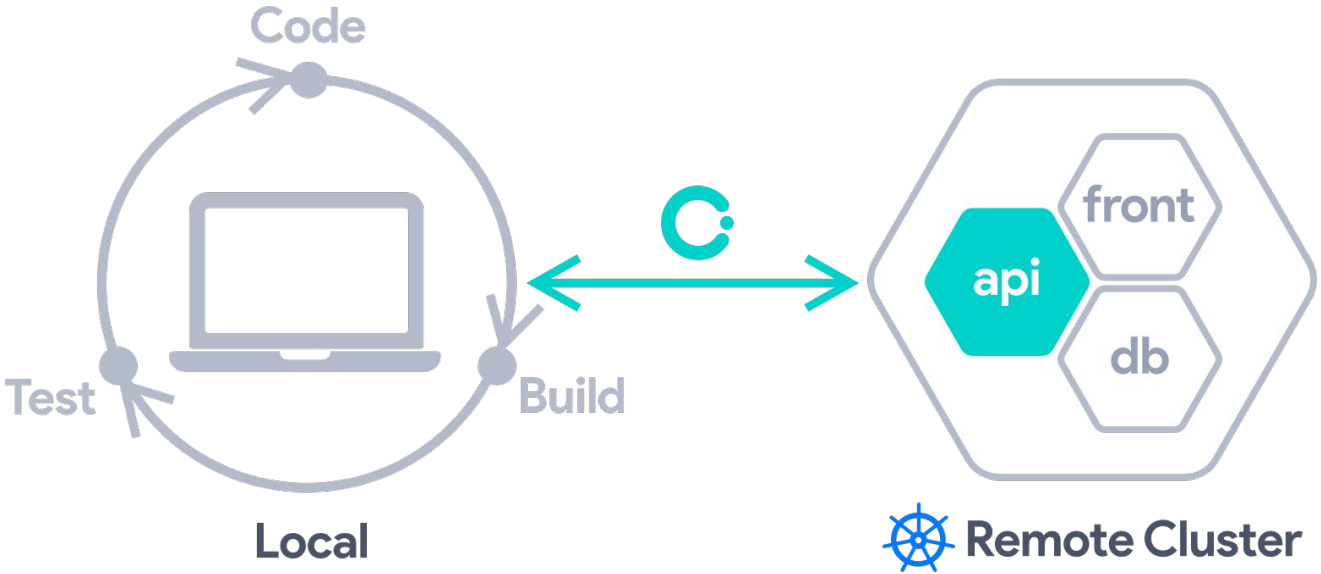
Your development environment is defined in a simple yaml manifest.

- Run `okteto init` to inspect your project and generate your own config file.
- Run `okteto up` to launch your development environment in seconds.

Demo time!

- https://github.com/rberrelleza/movies-demo

How does it work?


Code
Test
Build
Local
front
api
db
Remote Cluster

With okteto you get

- A manifest + container development environment
- Automatic file synchronization between local and remote env
- A remote terminal into your development environment
- Forward and Reverse port-forwarding
- Integration JetBrain and VSCode's remote development modes
- Single binary, works on linux, macos and windows

Developing in the same environment as your applications are going to run lets you go way faster.

You are not waiting on builds and deploys.

And you’re fully integrated from the very beginning.

You can leverage the entire platform as well as your stack’s toolkit.

Incremental builds.

Hot reloaders.

Debuggers!

Okteto works with anything that speaks Kubernetes

- Local clusters: Minikube, k3s, kind
- Remote clusters: EKS, GKE, AKS, Digital Ocean, Civo
- Kubernetes Manifests
- Helm
- Serverless functions
- ....

Interesting community use cases

- Core Kubernetes infrastructure and Operators
- Machine learning on GPU-heavy clusters
- Serverless Functions with OpenFaaS
- Deep integration with IDEs
- Develop directly in android/raspberry pi or other low power devices

Links!

- https://github.com/okteto/okteto
- https://github.com/okteto/samples
- https://marketplace.visualstudio.com/items?itemName=okteto.remote-kubernetes
- https://twitter.com/rberrelleza
- https://okteto.com/blog

Stars, feedback, issues and PR welcome!

https://github.com/okteto/okteto

Q&A

Thank you!

RAMIRO BERRELLEZA | @RBERRELLEZA

Okteto Stacks (sneak peak)

Docker-compose
for the Cloud Native World

Go from having to define your ingress, services, deployments, stateful sets, persistent volume claims ...

To this:

```yaml
name: voting-app
services:
  vote:
    public: true
    image: okteto/vote:stacks
    build: vote
    replicas: 2
    ports:
      - 8080
    resources:
      cpu: 100m
      memory: 128Mi

  redis:
    image: redis
    ports:
      - 6379
    resources:
      cpu: 100m
      memory: 128Mi
      storage: 1Gi
    volumes:
      - /data
```

Demo time!

- https://github.com/rberrelleza/movies-demo

We are just starting on this,
learn more at
https://github.com/okteto/stacks