

How to develop apps on Kubernetes and not die trying



*“It's almost become
boring to say that
Kubernetes has
become boring”*

Nov 2019
Maciej Szulik
k8s engineer at Red Hat
<http://bit.ly/k8s-boring>



DEVELOPERS



OPERATIONS



K8S ENGINEERS





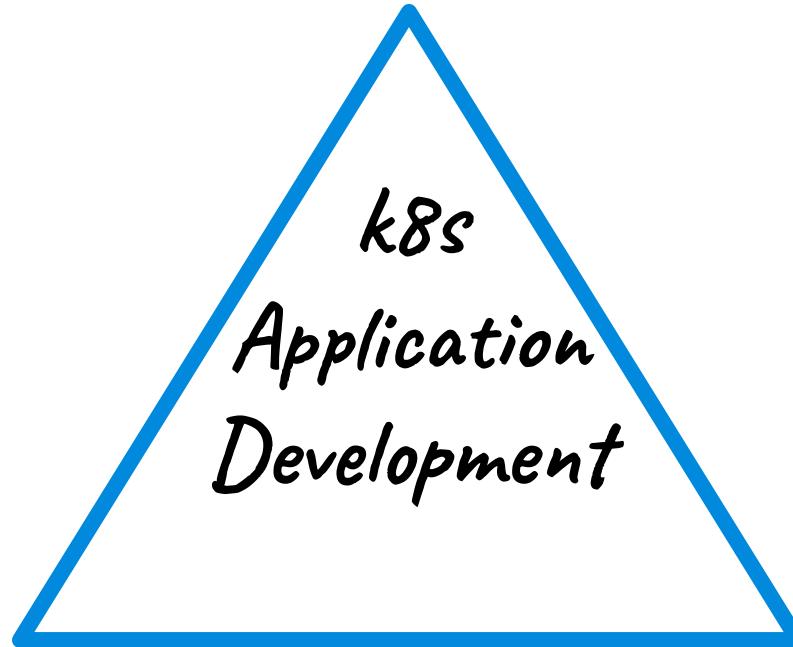
THE GATE
TIME





A photograph taken from a first-person perspective, looking down at a dark, textured asphalt surface. A thick, irregular pink chalk line starts near the bottom center and extends upwards and to the left. Within this line, the words "START HERE." are written in a bold, sans-serif font. The person's feet, wearing dark lace-up shoes, are positioned directly beneath the end of the chalk line. To the left of the line, there is a small, crumpled white piece of paper. The overall composition suggests a starting point or a call to action.

START
HERE.



The 3 angles of k8s application development

A close-up photograph of an open book lying flat. The pages are numerous, aged, and yellowed, fanning out from the center. The book is bound in dark blue leather covers visible at the edges.

Definitions



The word "APPLICATION" is spelled out in black letters on white rectangular tiles, arranged horizontally on a dark brown wooden surface with visible grain. The tiles are slightly angled, creating a sense of depth.

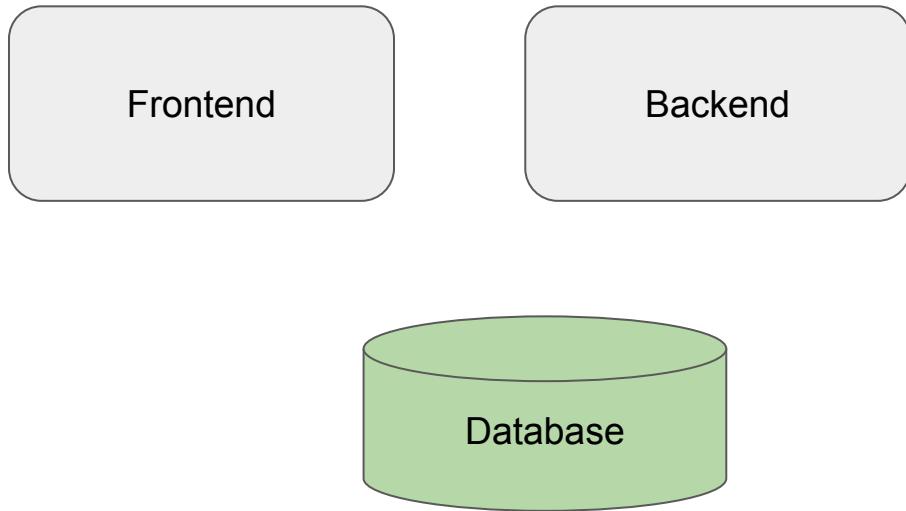
APP LIC ATION

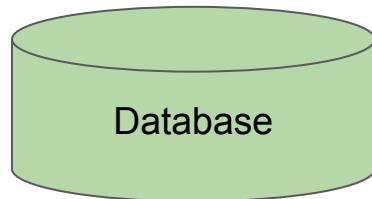
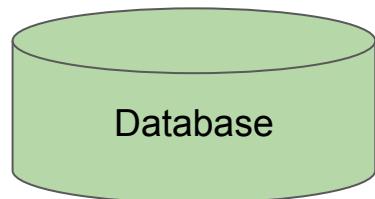
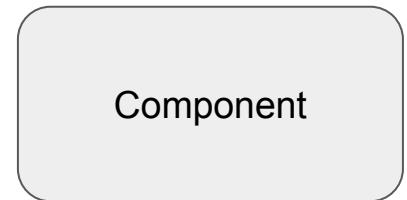
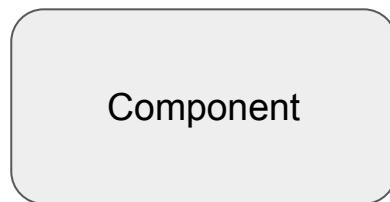
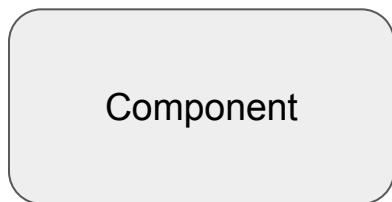
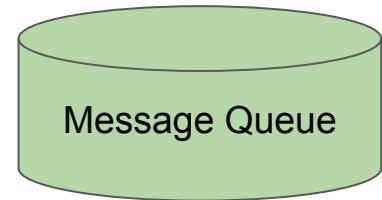
What is an application?

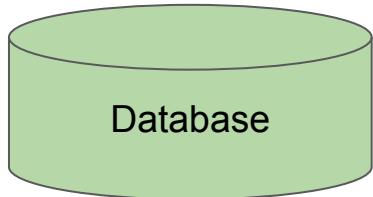
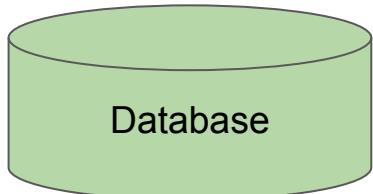
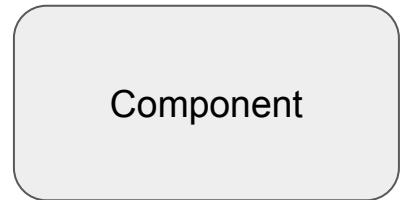
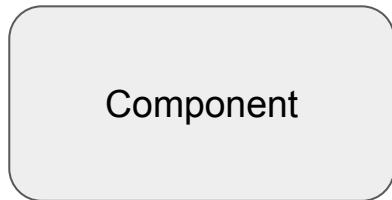
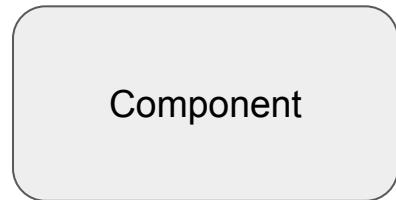
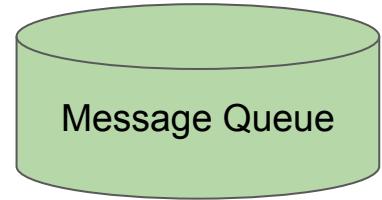
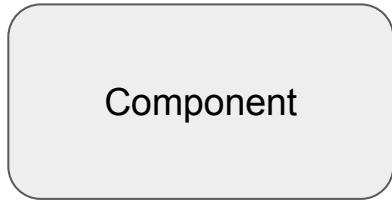
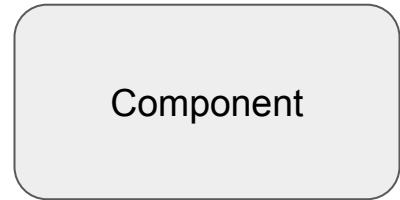
Monolith

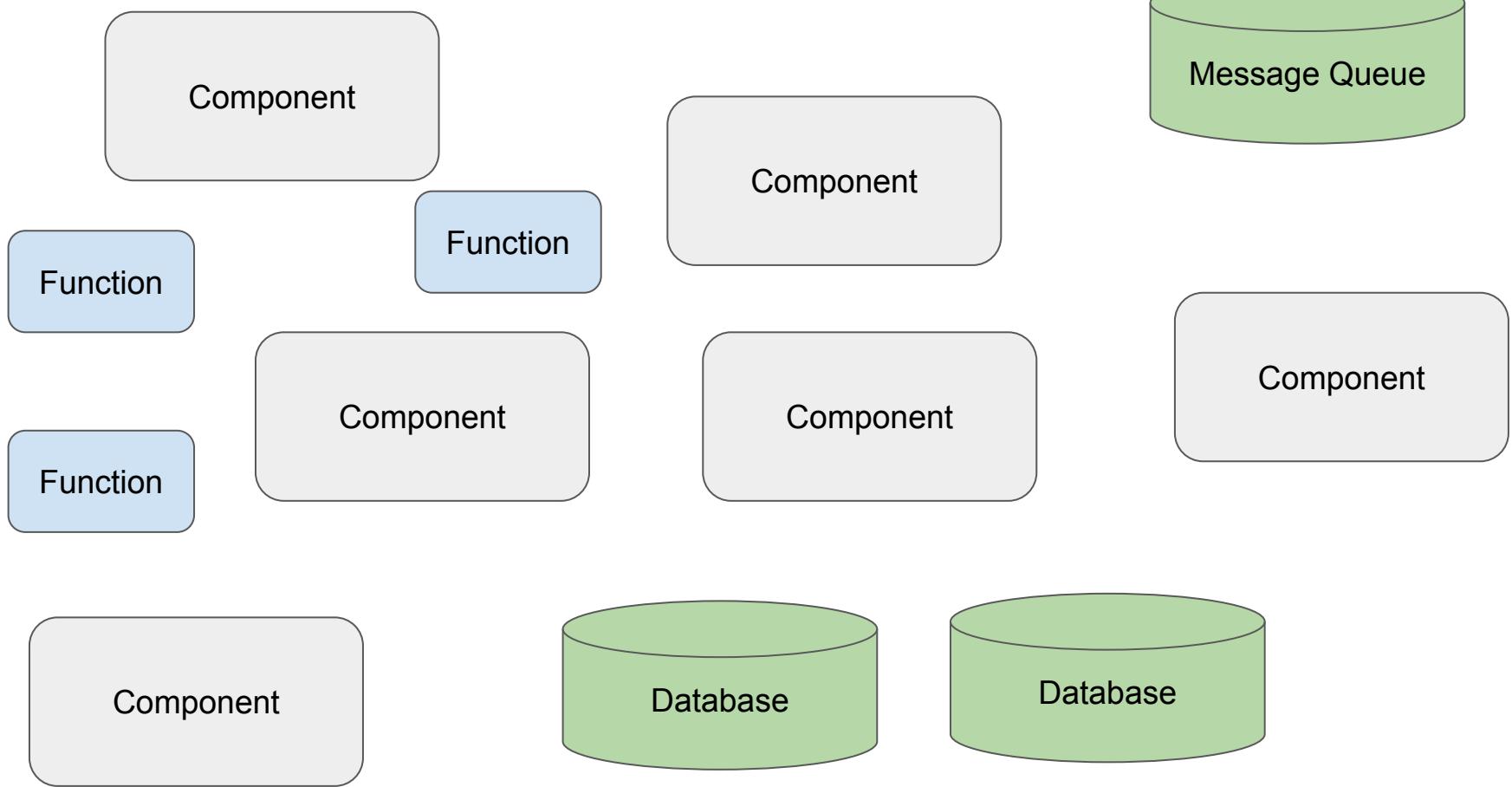
Frontend

Backend









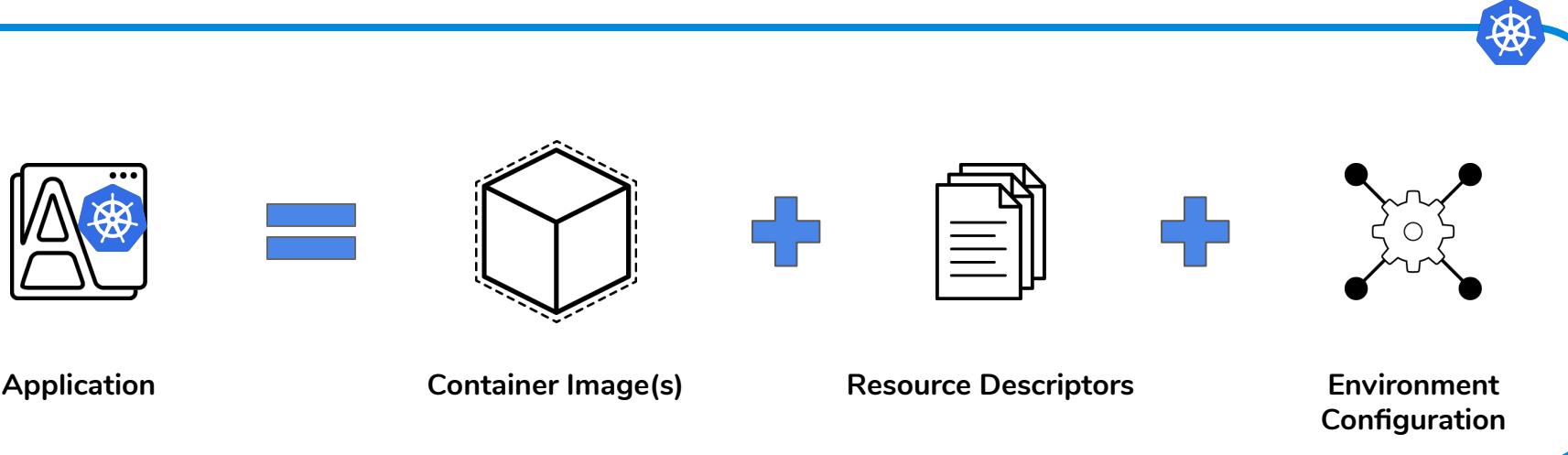
“An application is any software component, or group of components, that is designed for the end user”

“An application is any software component, or group of components, that is designed for the end user”

- Complete and fully functional
- Does have a version number representing it
- Multiple instances with different configuration can exist
- Can be easily installed



Logical vs Physical views

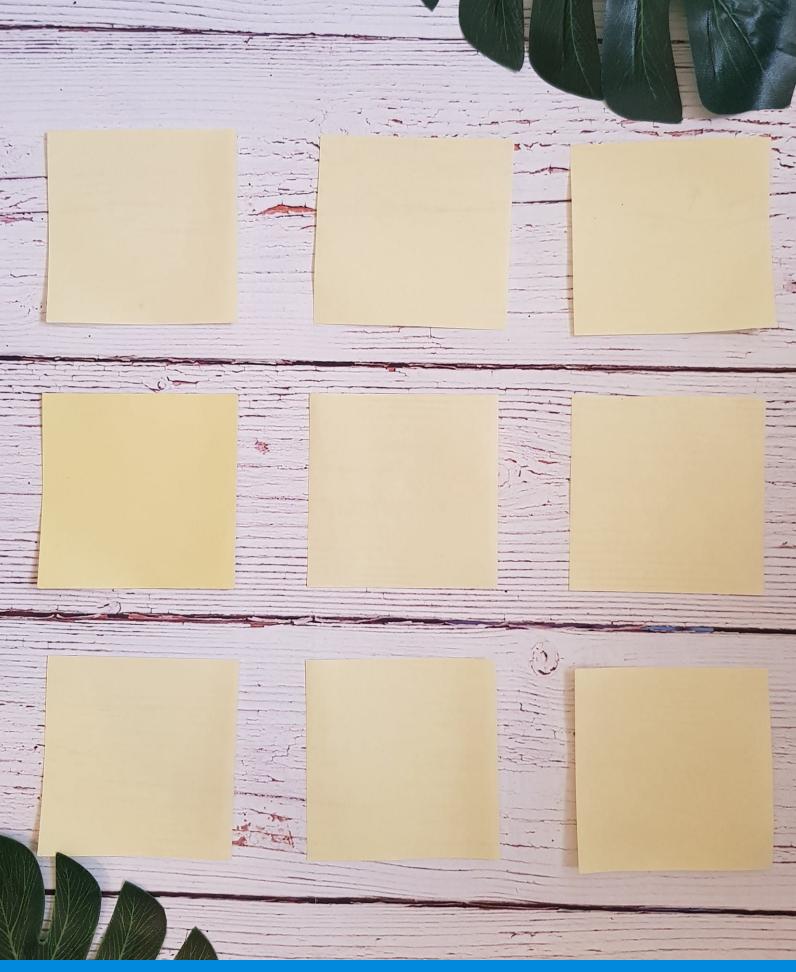


k8s application physical view

*Application
Definition*

k8s

*Application
Development*



```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: myapp
    app.kubernetes.io/instance: myapp-abcxyz
    app.kubernetes.io/version: "1.0.0"
    app.kubernetes.io/managed-by: kapp
    app.kubernetes.io/component: server
    app.kubernetes.io/part-of: myapp
```

Kubernetes recommended application labels

```
apiVersion: app.k8s.io/v1beta1
kind: Application
metadata:
  name: "wordpress-01"
  labels:
    app.kubernetes.io/name: "wordpress-01"
    app.kubernetes.io/version: "3"
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: "wordpress-01"
  componentKinds:
    - group: core
      kind: Service
    - group: apps
      kind: Deployment
    - group: apps
      kind: StatefulSet
  assemblyPhase: "Pending"
...

```

```
...
descriptor:
  version: "4.9.4"
  description: "WordPress is oss you can use to create a
beautiful website, blog, or app."
  icons:
    - src: "https://example.com/wordpress.png"
      type: "image/png"
  type: "wordpress"
  maintainers:
    - name: Kenneth Owens
      email: kow3ns@github.com
  owners:
    - "Kenneth Owens kow3ns@github.com"
  keywords:
    - "cms"
    - "blog"
    - "wordpress"
  links:
    - description: About
      url: "https://wordpress.org/"
    - description: Web Server Dashboard
      url: "https://metrics/internal/wordpress-01/web-app"
    - description: Mysql Dashboard
      url: "https://metrics/internal/wordpress-01/mysql"
```

Application CRD

```
apiVersion: etcd.database.coreos.com/v1beta2
kind: EtcdCluster
metadata:
  name: example
spec:
  size: 3
  version: 3.2.13
```

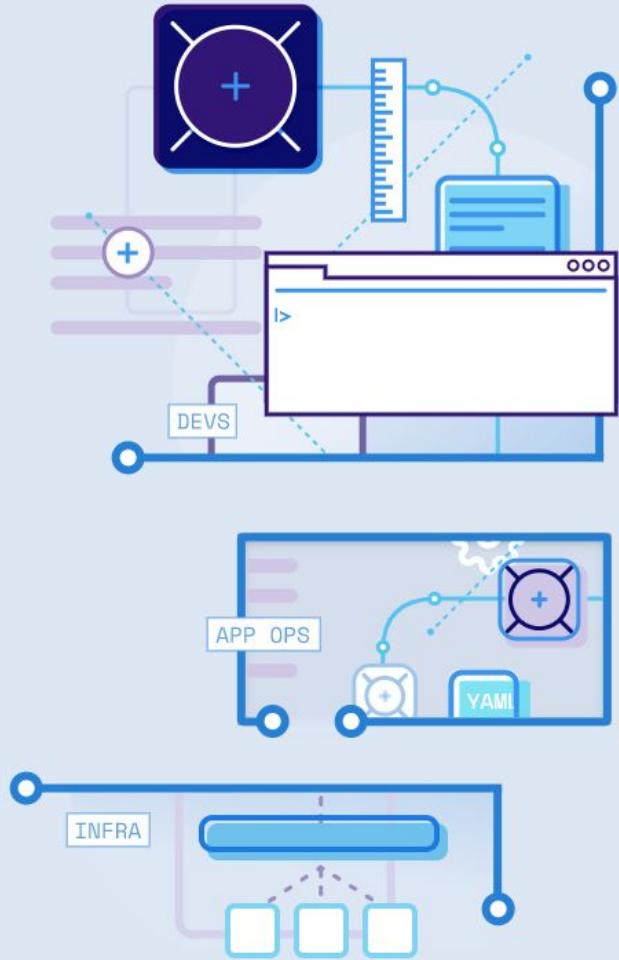
Custom CRD



“A spec for
packaging
distributed apps”

cnab.io

CNAB



“A team-centric
standard for building
cloud native apps”

oam.dev

OAM



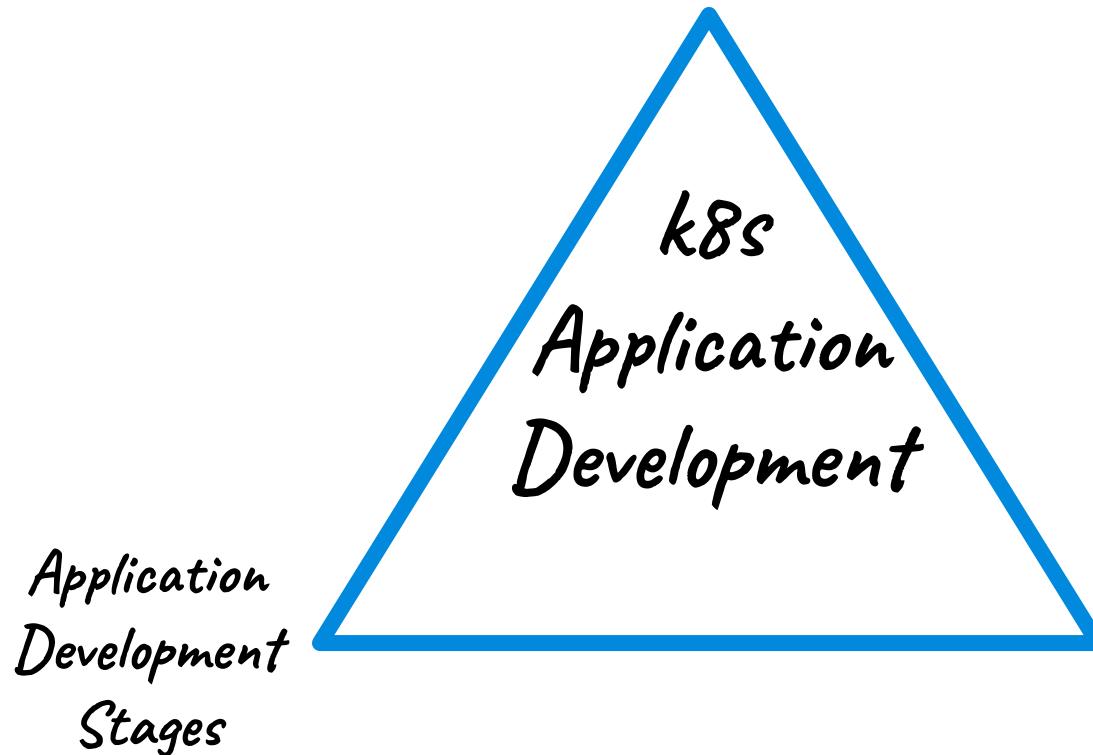
```
apiVersion: core.oam.dev/v1alpha1
kind: ComponentSchematic
metadata:
  name: alpine-forever-v1
spec:
  workloadType: core.oam.dev/v1alpha1.SingletonServer
  parameters:
    - name: message
      type: string
      required: false
    - name: unused_integer
      type: number
      required: false
      default: 5678
  containers:
    - name: runner
      image: technosophos/alpine-forever:latest
      env:
        - name: FOO
          value: bar
          fromParam: message
        - name: UNUSED
          value: "1234"
          fromParam: unused_integer
```

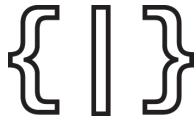


A Kubernetes Implementation of the
Open Application Model



rudr (OAM Implementation)

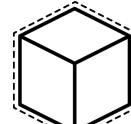




Source code
authoring



Descriptors
authoring



Packaging



Application



Deployment

k8s application development stages

Repeat!!!



kubectx + kubens



OCTANT



KiTS
Kubernetes Tools



Configure Unify Execute

+

Kaniko



Buildpacks.io

Source-To-Image (S2I)

k14s

Kubernetes Tools that follow Unix philosophy to be simple, and composable.

[Github](#)

[#k14s in Kubernetes slack](#)

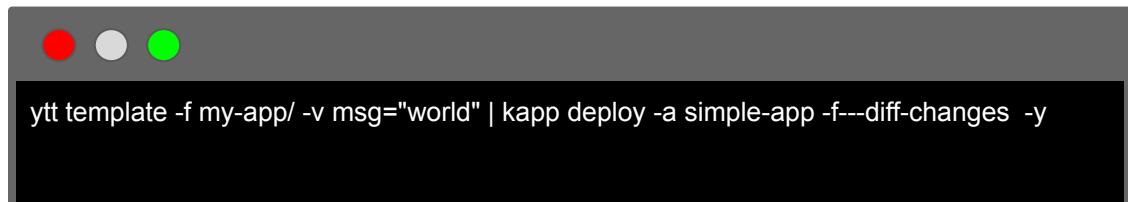
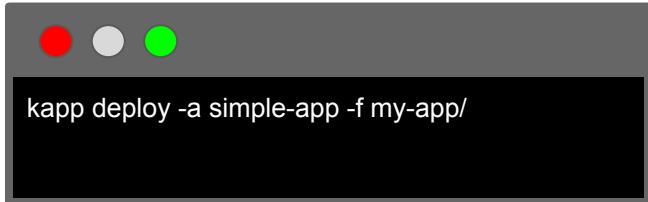
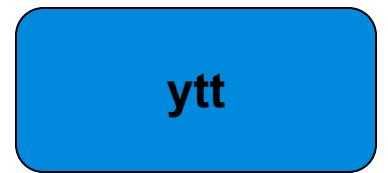
Tools

- **ytt** #template, #overlay
 - is a YAML templating and overlay tool that works on YAML structure instead of text
- **kbld** #build, #img-tag-to-digest, #registry-relocate
 - helps working with container images used in Kubernetes configuration files
- **kapp** #deploy, #diff-apply
 - is a deployment tool with a concept of "Kubernetes application" – a set of resources with the same label
- **imgpkg** #package
 - is for storing application configuration files in Docker/OCI registries
- **kapp-controller** #gitops
 - introduces App CRD to run templating plus deployment workflow in-cluster (via ytt and kapp)
- **vendir**
 - allows to declaratively state what files should be in a directory
- **kwt**
 - is Kubernetes Workstation Tools CLI which provides convenient networking tools and development workspaces

kapp



```
kapp deploy -a simple-app -f my-app/
```



kblD



ytt



kapp

A dark gray rectangular box representing a terminal window, featuring three colored circular icons (red, white, green) at the top.

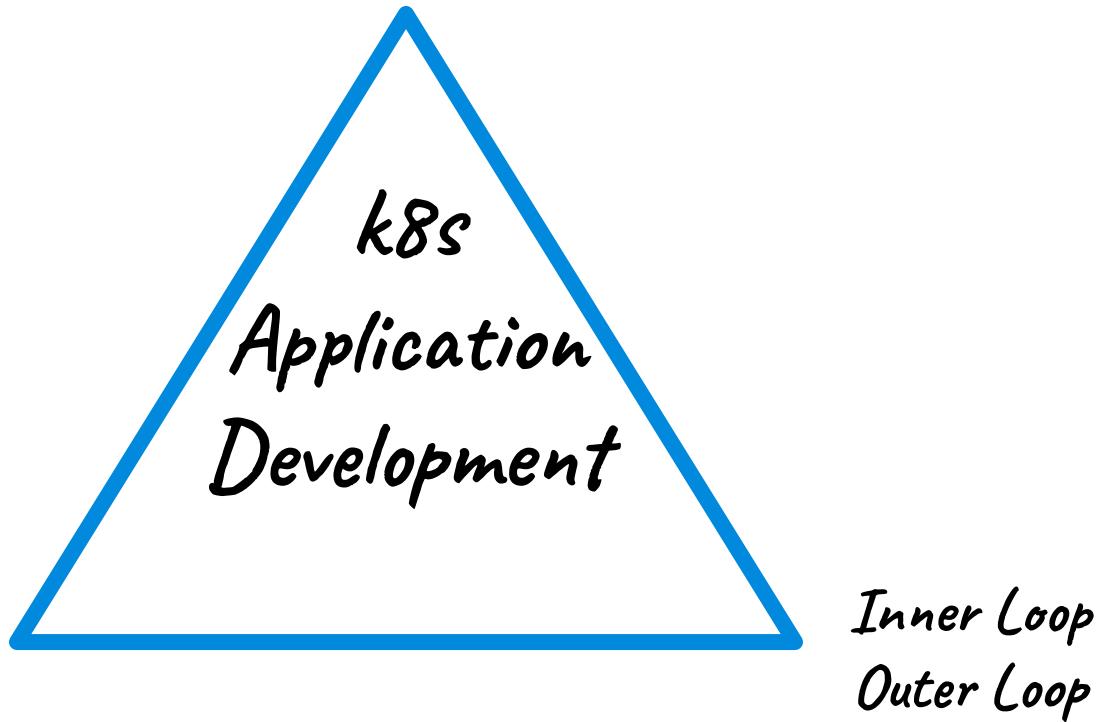
```
kapp deploy -a simple-app -f my-app/
```

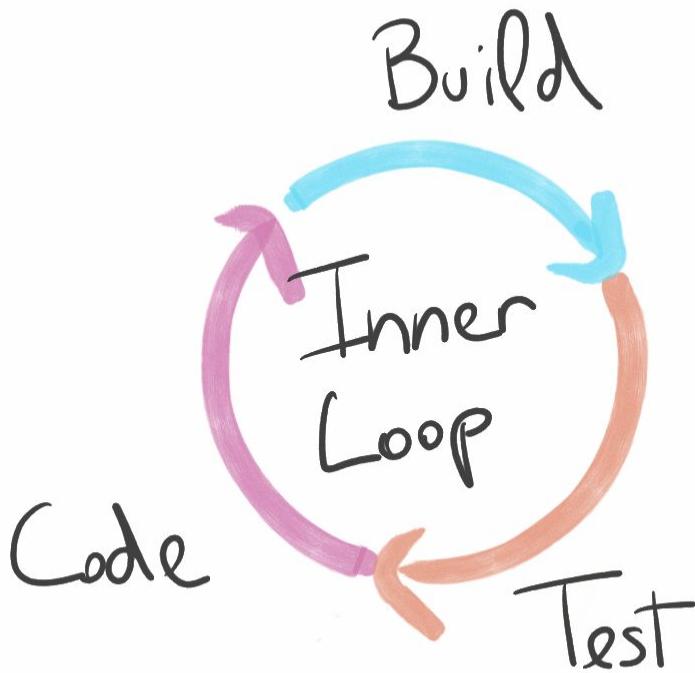
A dark gray rectangular box representing a terminal window, featuring three colored circular icons (red, white, green) at the top.

```
ytt template -f my-app/ -v msg="world" | kapp deploy -a simple-app -f--diff-changes -y
```

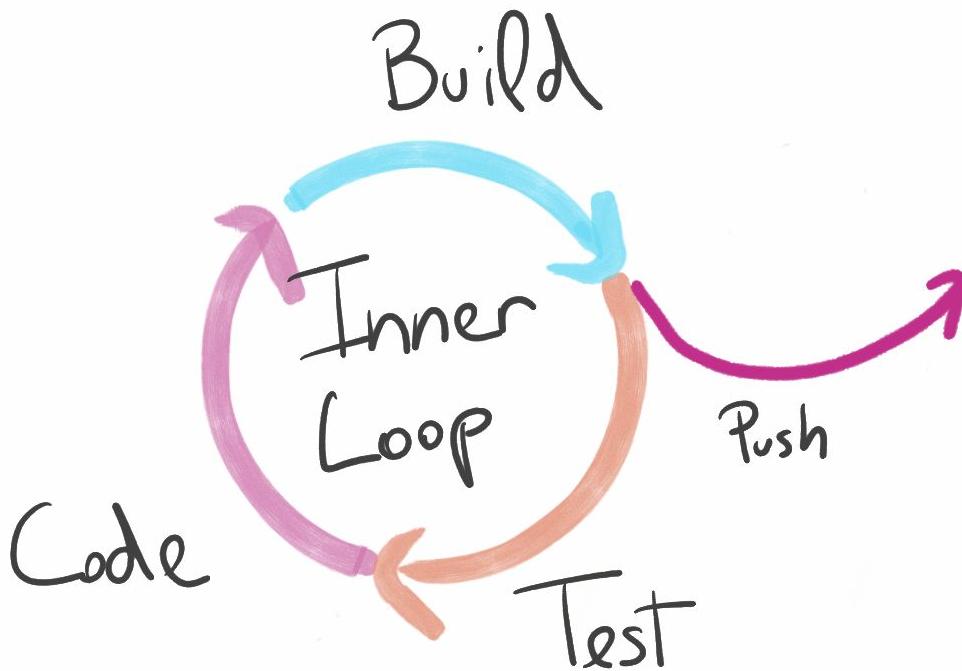
A dark gray rectangular box representing a terminal window, featuring three colored circular icons (red, white, green) at the top.

```
ytt template -f my-app/ -f my-app-overlay/custom-scale.yml -v msg="world" | kblD -f- | kapp deploy -a simple-app -f- --diff-changes -y
```

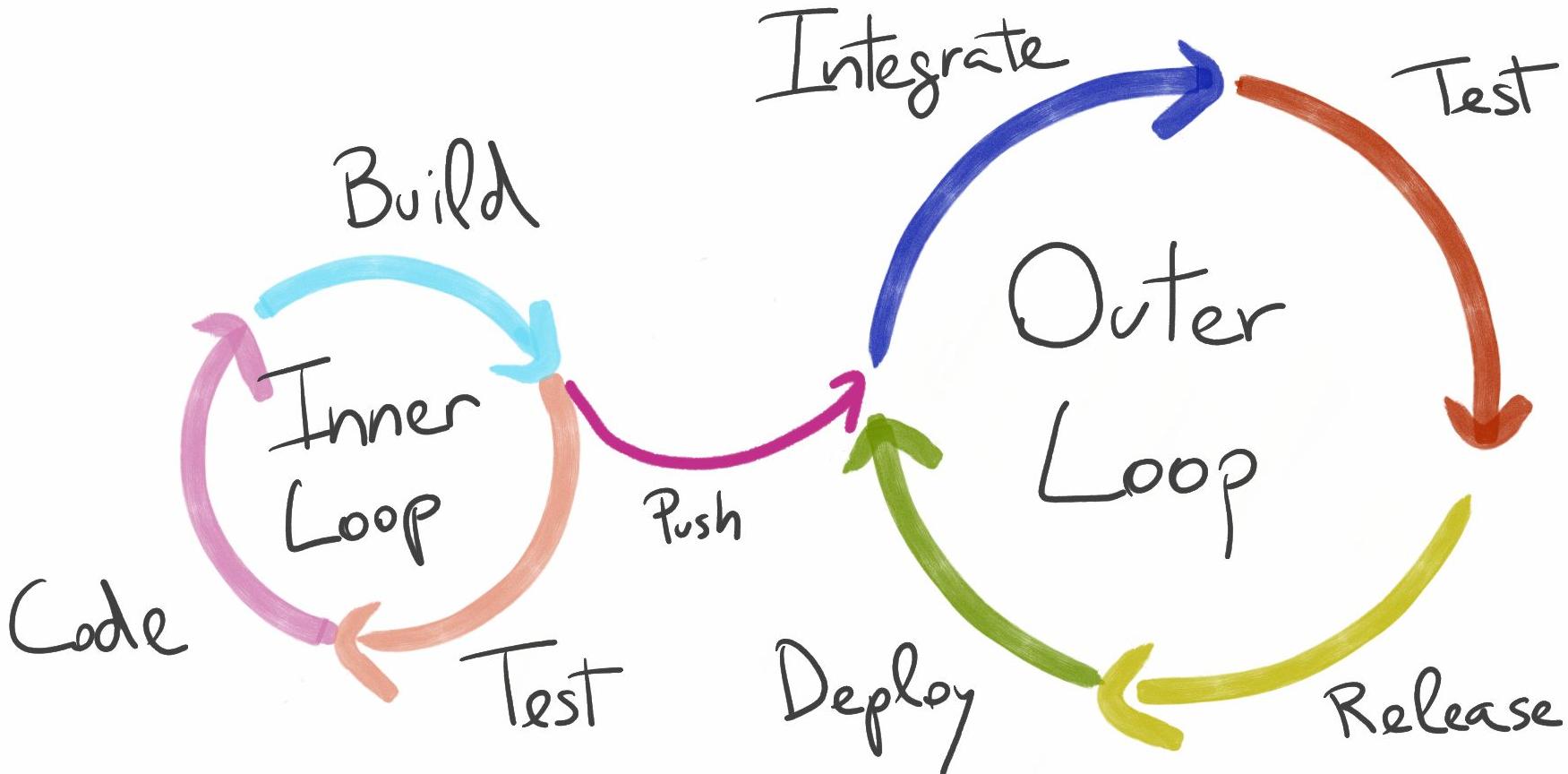




Inner Loop / Outer Loop

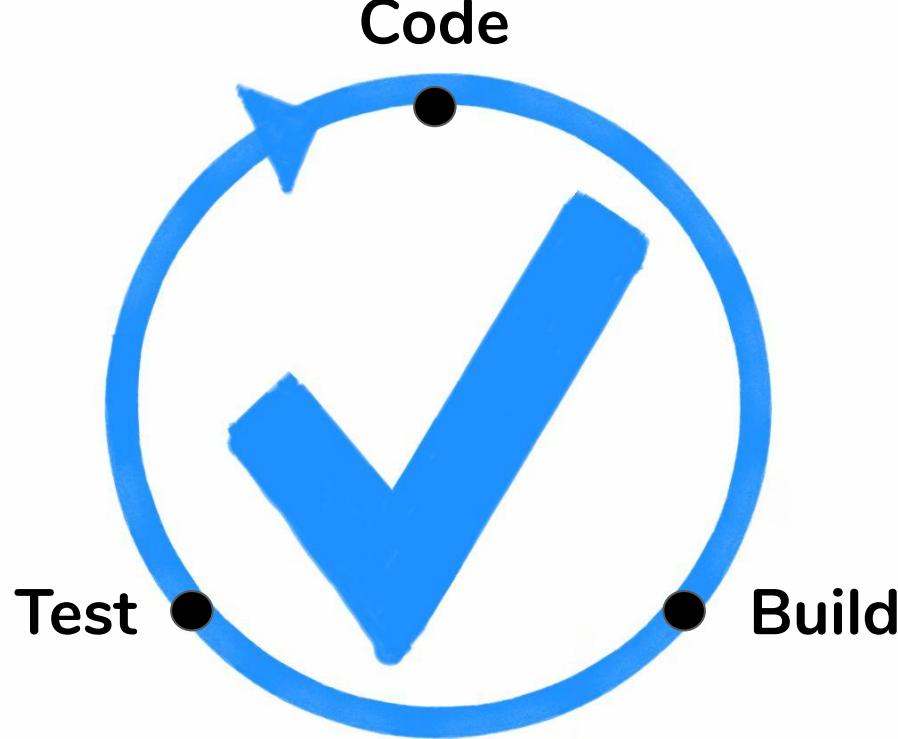


Inner Loop / Outer Loop

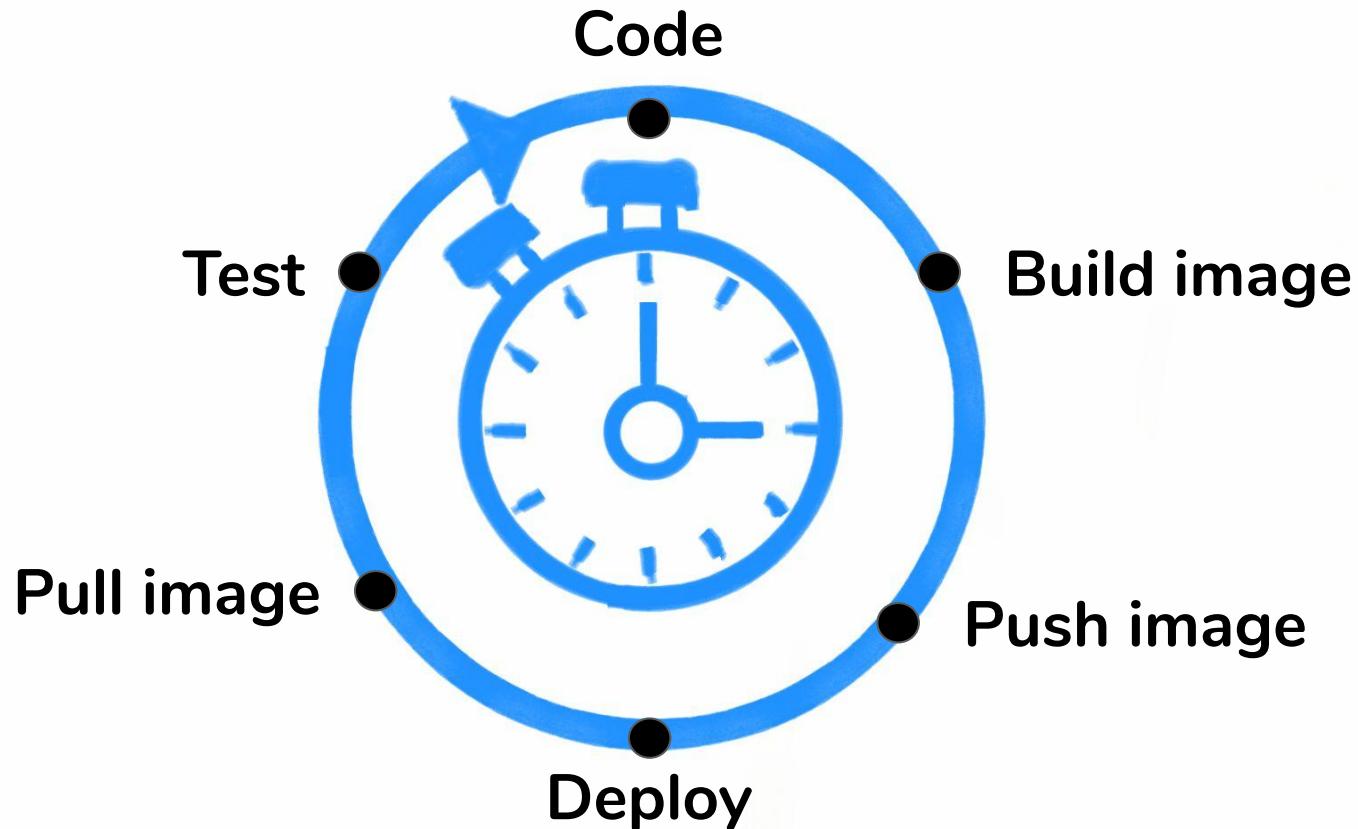


Inner Loop / Outer Loop

Code



Inner Loop

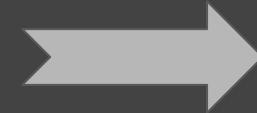


Inner Loop with containers

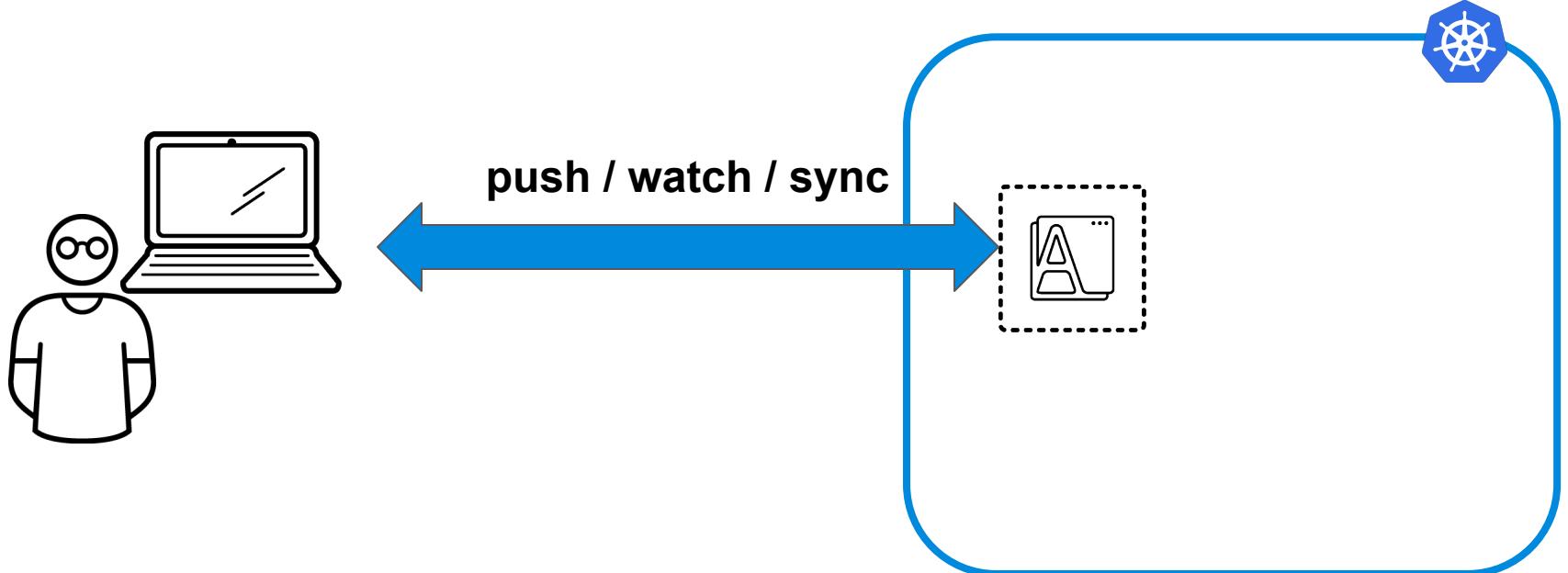
CTRL

+

S



Fast Inner Loop with Containers



Inner Loop with containers



openshift / odo

conclusion



Simple tools vs Power tools





Do we want Voodoo to happen?



“Consistency
as a
feature”



*“It's almost become
boring to say that
developing
applications on
Kubernetes has
become **easy**”*

Someday in the future
Jorge Morales
Developer Advocate

bit.ly/develop-k8s-app-talk

Jorge Morales Pou

Cloud Native Developer Advocate @VMware



<https://jorgemoral.es>



jorgemoralespou



@jorgemoralespou