

# Quick Response Kubernetes Dev Workflow

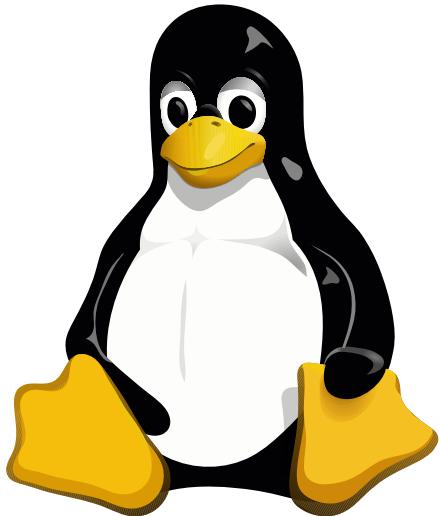
Immanuel Sims  
BED-Con 2023, Berlin

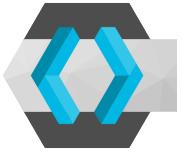


# Agenda

- ▶ Motivation
- ▶ Available Tools
- ▶ Solution
  - Plan
  - Execution with Maven
  - Execution with Gradle
- ▶ Reviewing our Achievements

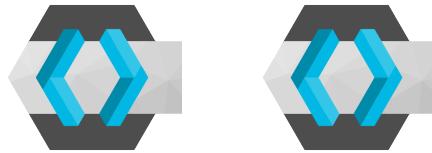
# Motivation





Keycloak

# Motivation

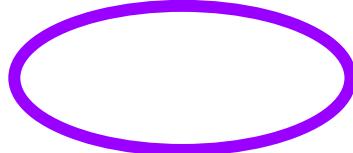




...



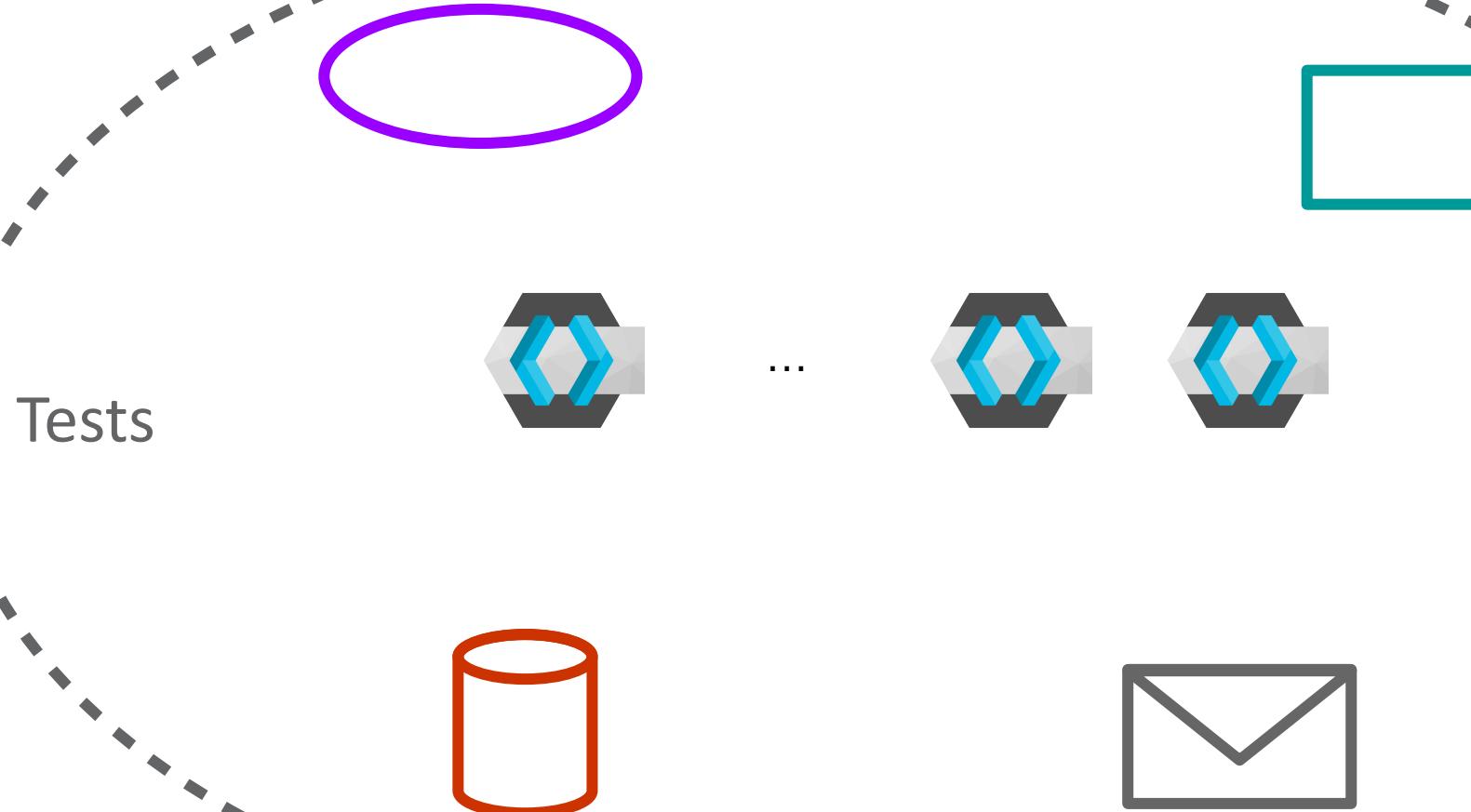
# Motivation



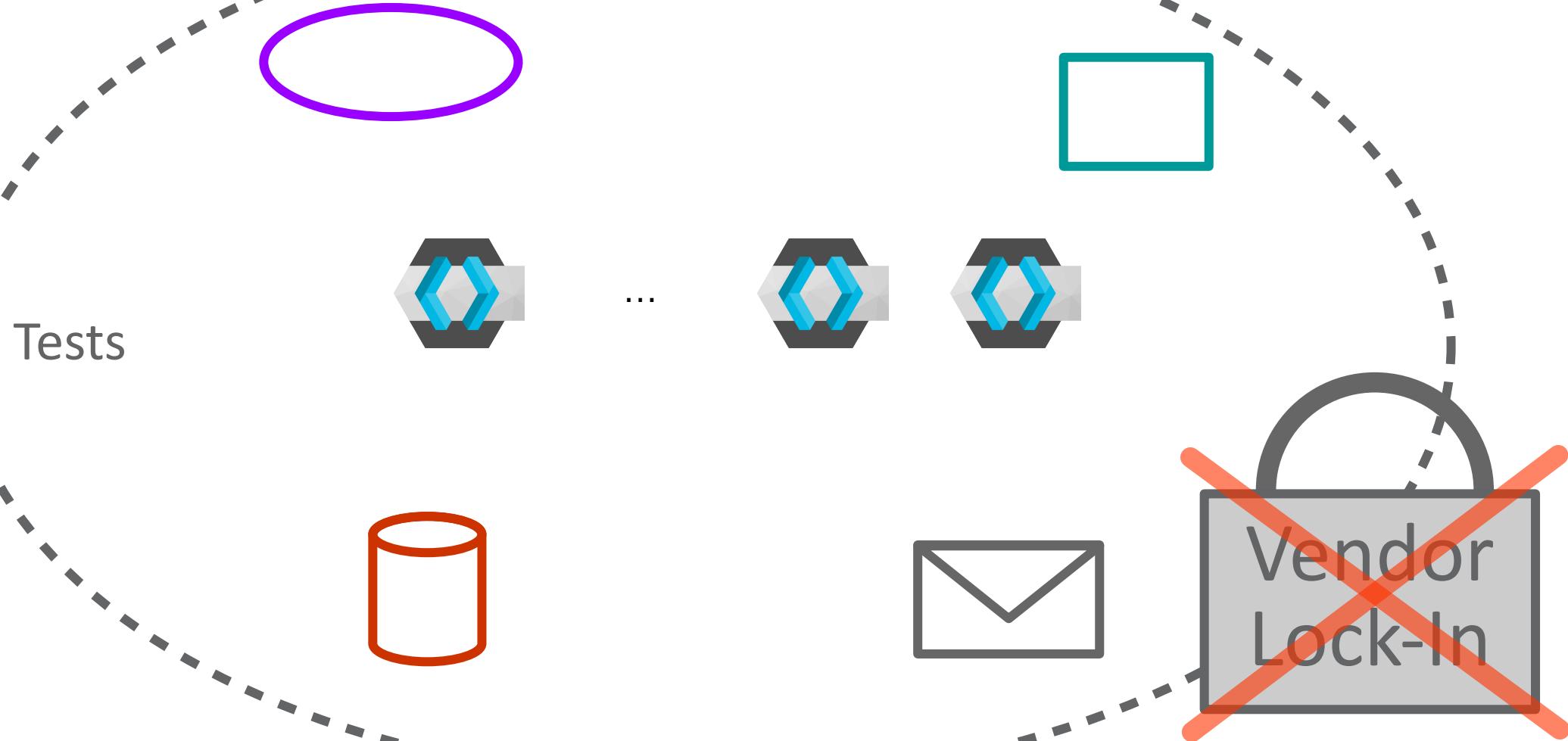
...



# Motivation



# Motivation



# Requirements

- ▶ easy to install
- ▶ easy to scale
- ▶ no vendor lock in
- ▶ quick & realistic E2E testing

# Requirements

- ▶ easy to install → ▶ container
- ▶ easy to scale → ▶ container orchestration
- ▶ no vendor lock in → ▶ K8S
- ▶ quick & realistic E2E testing → ▶ K8S for each dev

# Requirements

- ▶ easy to install → ▶ container
- ▶ easy to scale → ▶ container orchestration
- ▶ no vendor lock in → ▶ K8S
- ▶ quick & realistic E2E testing → ▶ K8S for each dev
- ▶ want to use CRDs → ▶ definitely K8S for each dev

# Tooling Options



minikube



S K A F F O L D



Google Cloud Platform



K3S

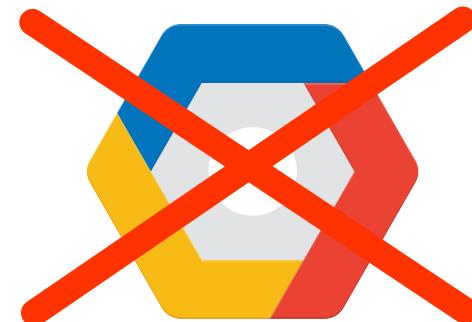
# Tooling Options



minikube



S K A F F O L D



Google Cloud Platform

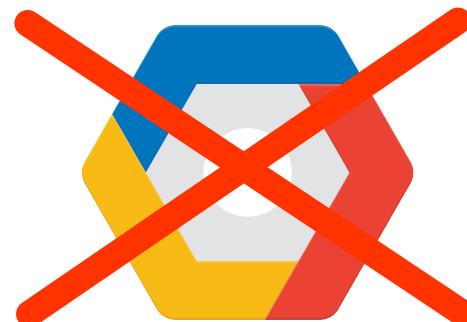


K3S

# Tooling Options



minikube



Google Cloud Platform



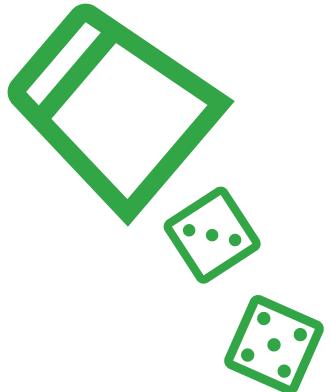
**K3S**



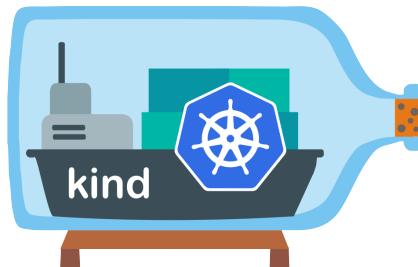
# Tooling Options



minikube

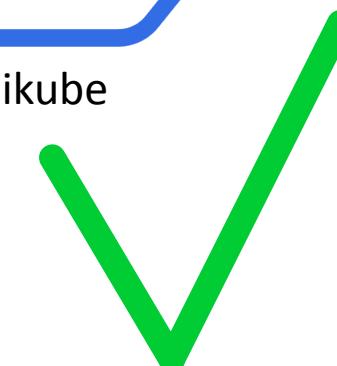


**K3S**



# Tooling Options

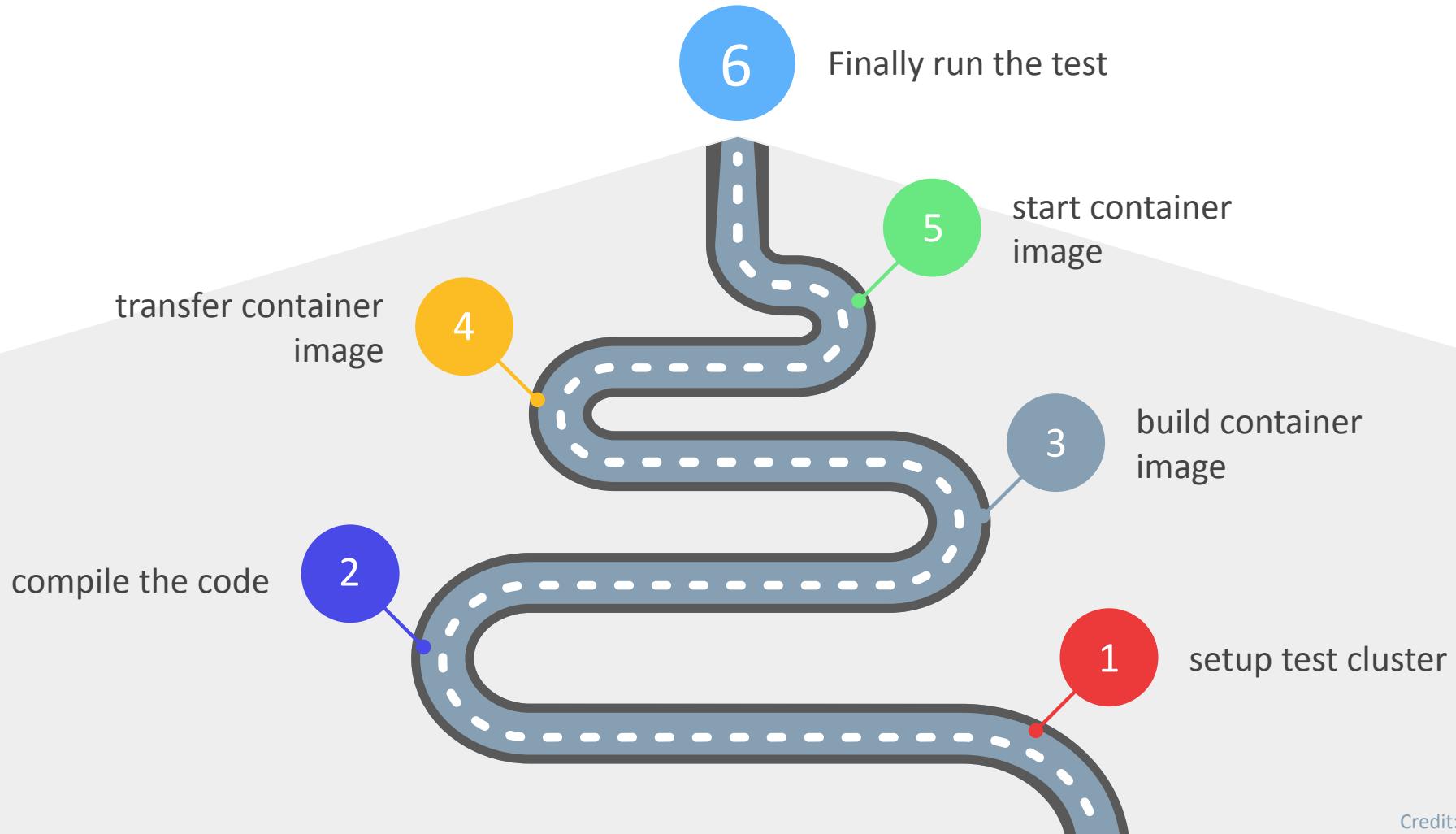
- ▶ driver options
- ▶ configurable k8s version
- ▶ persistent storage
- ▶ ...



**K3S**

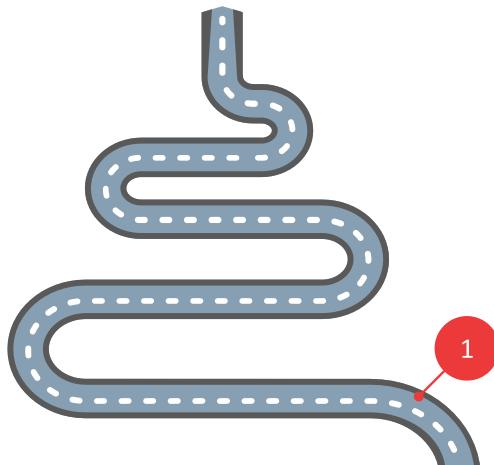
# What needs to be done?

# Running a System Test



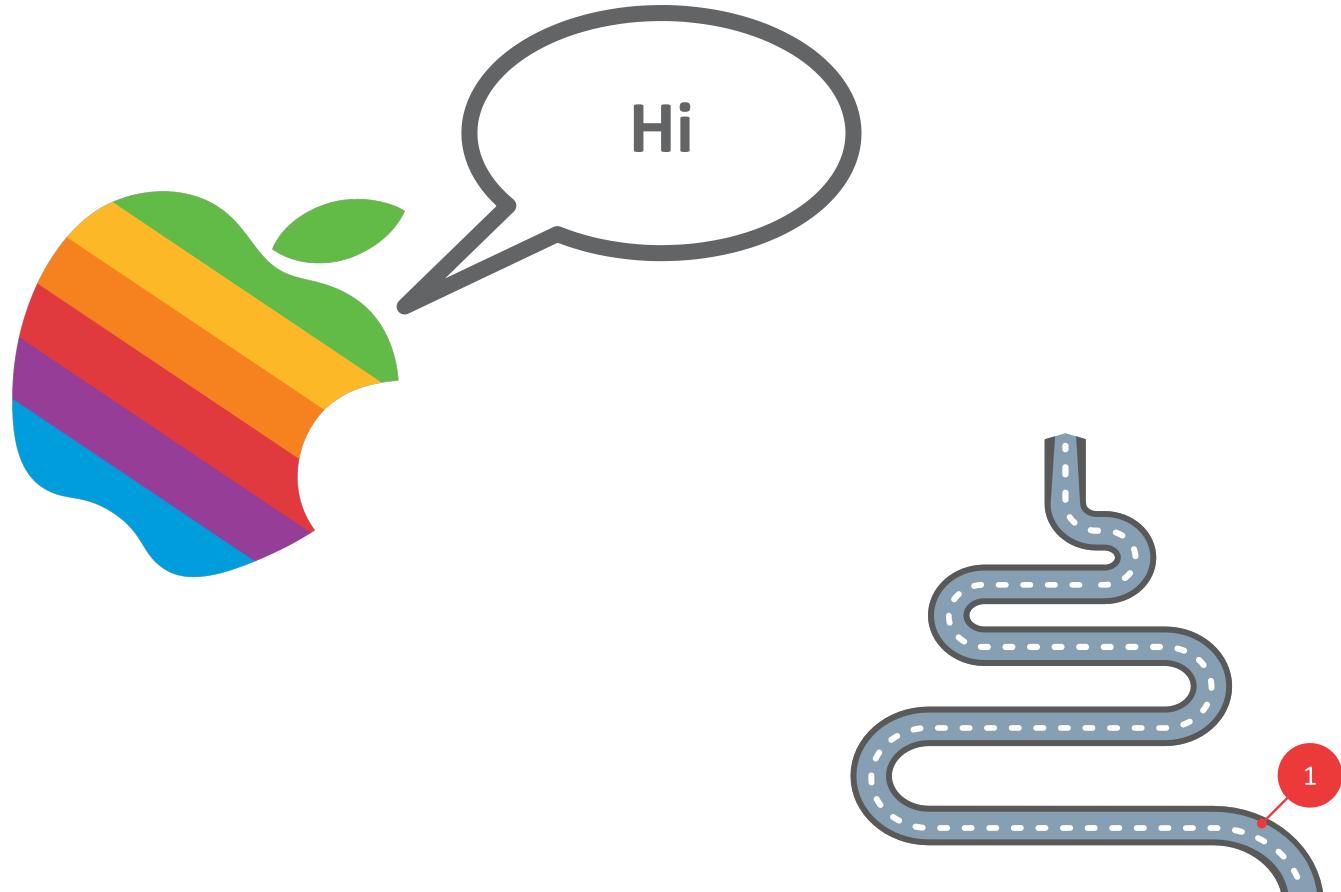
# 1) set up access to test cluster

- ▶ »minikube start
- ▶ options for
  - --memory
  - --cpus
  - --container-runtime
  - --kubernetes-version



# 1) set up access to test cluster

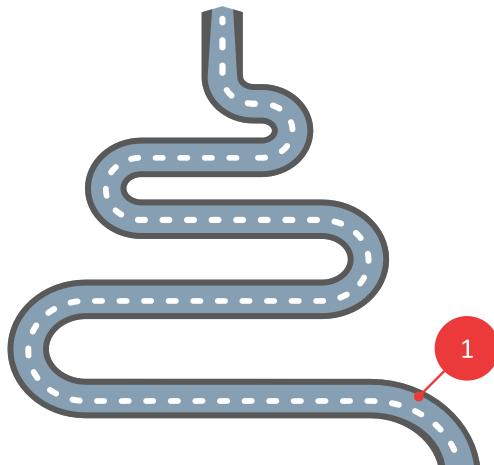
- ▶ »minikube start
- ▶ options for
  - --memory
  - --cpus
  - --container-runtime
  - --kubernetes-version



# 1) set up access to test cluster

- ▶ »minikube start
- ▶ options for
  - --memory → on MacOs beware docker machine memory
  - --cpus
  - --container-runtime
  - --kubernetes-version
- ▶ more options because apple
  - --driver=docker
  - --ports [...] ← Forward Node Ports

All Happy:



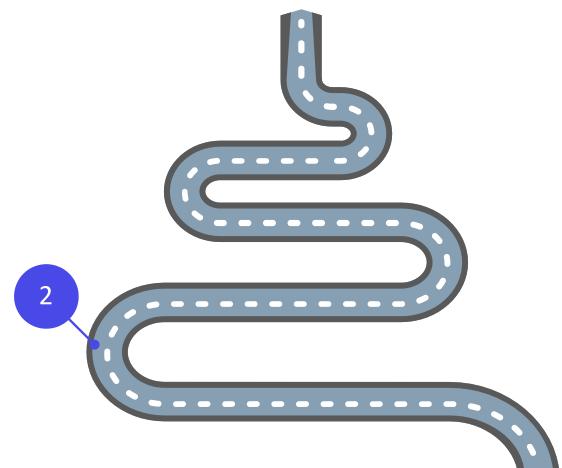
## 2) compile code

- ▶ business as usual



imgflip.com

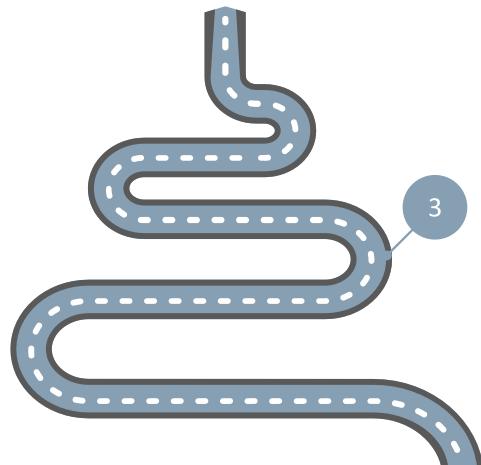
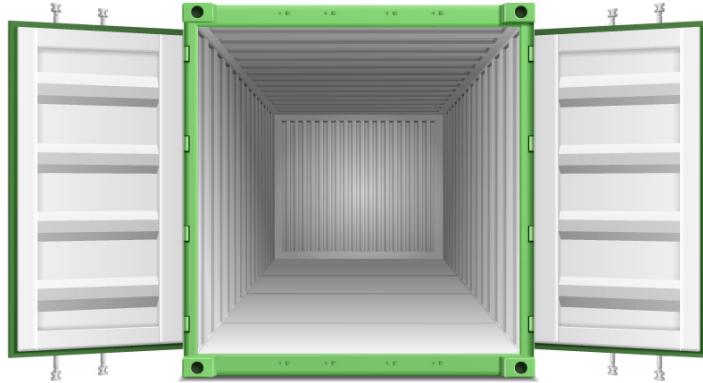
Credit: Imageflip



### 3) build container image

► Options:

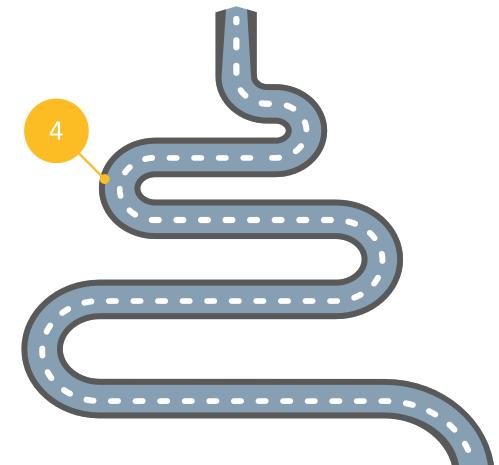
- a) » docker build
- b) jib



## 4) transfer container image

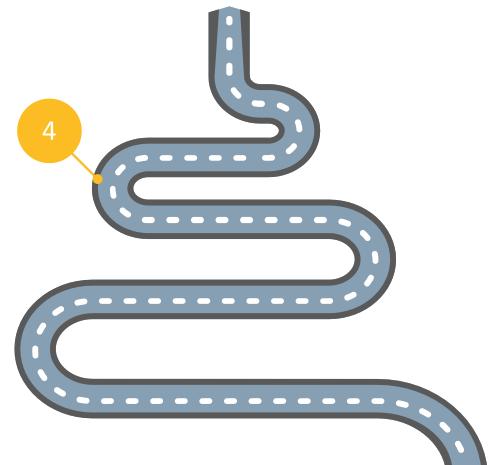


Credit: vecstock on Freepik



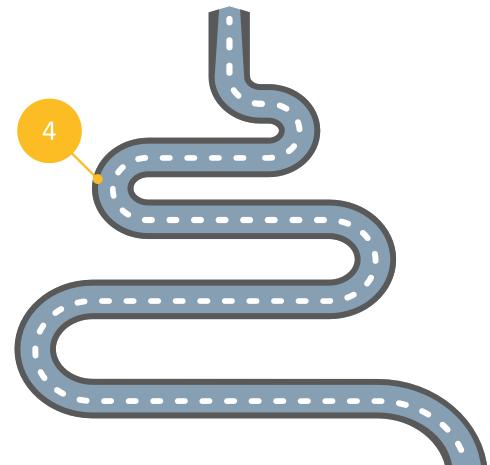
## 4) transfer container image

- ▶ Options:
  - a) » docker push



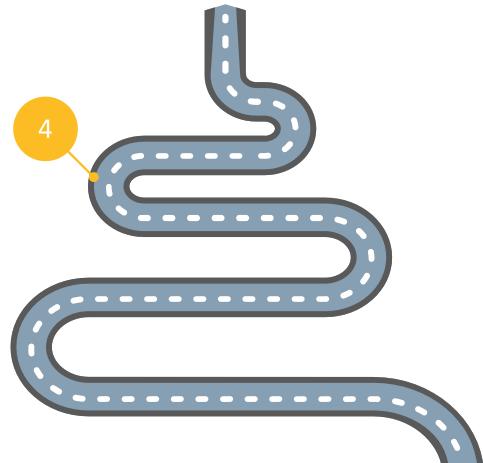
## 4) transfer container image

- ▶ Options:
  - a) » docker push
  - b) » minikube load



## 4) transfer container image

- ▶ Options:
  - a) » docker push
  - b) » minikube load
  - c) » minikube docker-env with
    - » docker build
    - » jib:dockerBuild



## 4) transfer container image

- 1. bad because net roundtrips
- 2. bad because confusion in registry

### ► Options:

- a) » `docker push`
- b) » `minikube load`
- c) » `minikube docker-env` with
  - » `docker build`
  - » `jib:dockerBuild`

bad because rare bugs

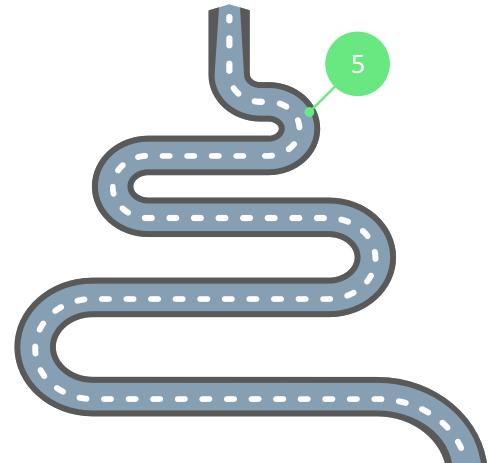
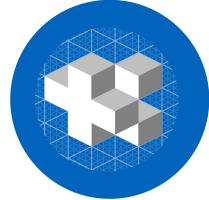
good because it's flexible

can't do cache optimizations

good because it can join 2 steps

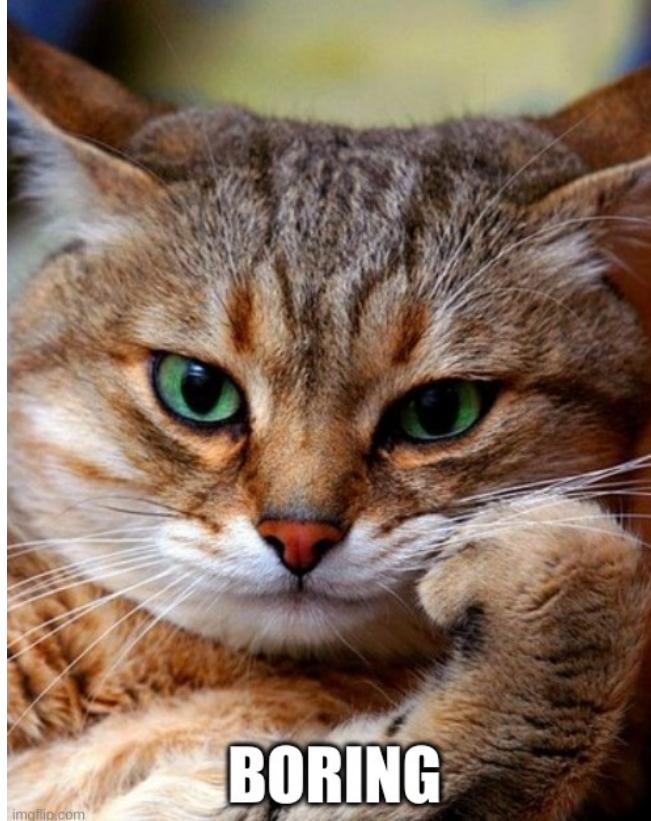
## 5) start container image

- ▶ make use of your config management
  - » kubectl apply & kubectl rollout status
  - » helm
  - ...
- ▶ use image pull policy: Never

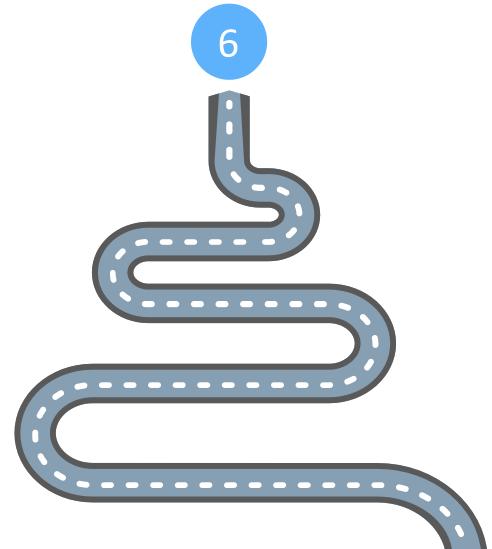


## 6) actually run systemtest

- ▶ business as usual



Credit: Imageflip



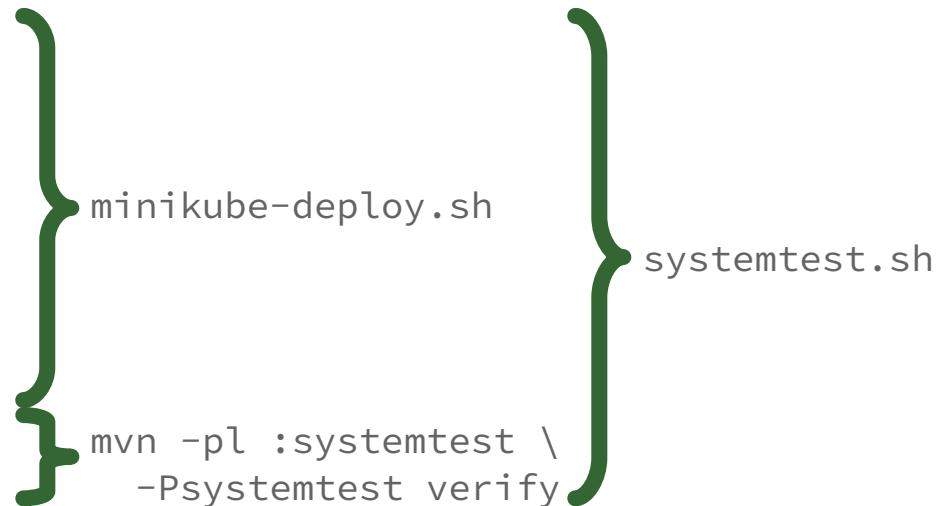
How could this look like with  
maven ?

- 1) setup test cluster
- 2) compile the code
- 3) build container image
- 4) transfer container image
- 5) start container image
- 6) run the test



systemtest.sh

- 1) setup test cluster
- 2) compile the code
- 3) build container image
- 4) transfer container image
- 5) start container image
- 6) run the test

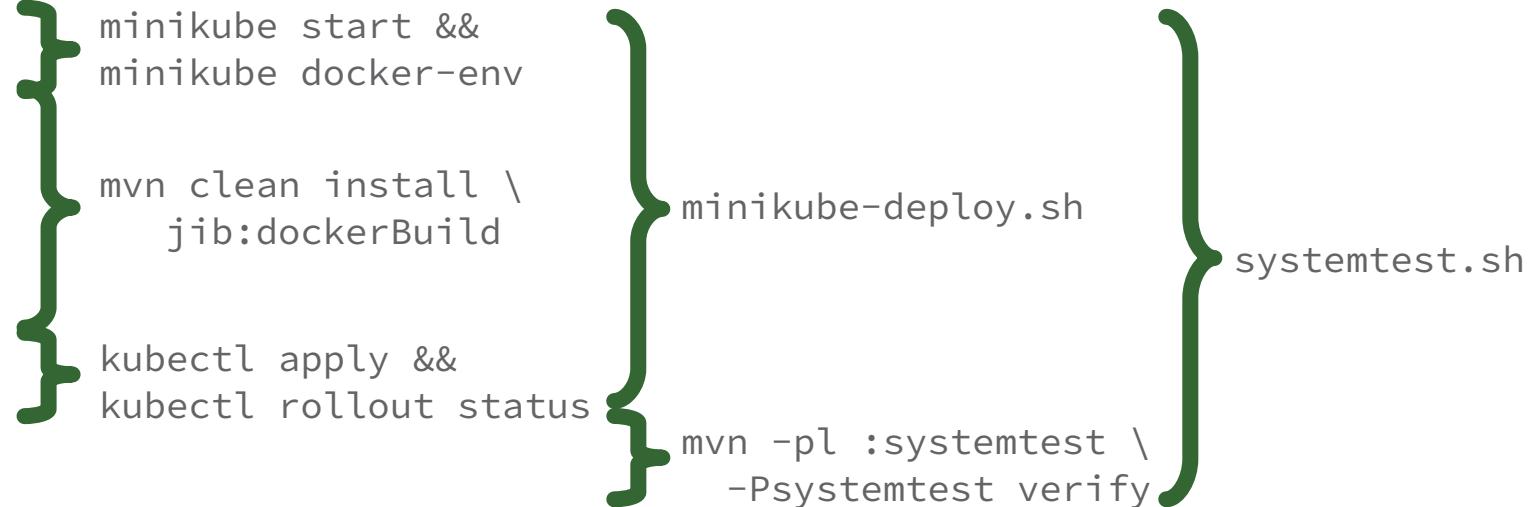


```
mvn -pl :systemtest \
      -Psystemtest verify
```

minikube-deploy.sh

systemtest.sh

- 1) setup test cluster
- 2) compile the code
- 3) build container image
- 4) transfer container image
- 5) start container image
- 6) run the test



```
minikube start &&
minikube docker-env

mvn clean install \
jib:dockerBuild

kubectl apply &&
kubectl rollout status
```

minikube-deploy.sh

mvn -pl :systemtest \
-Psystemtest verify

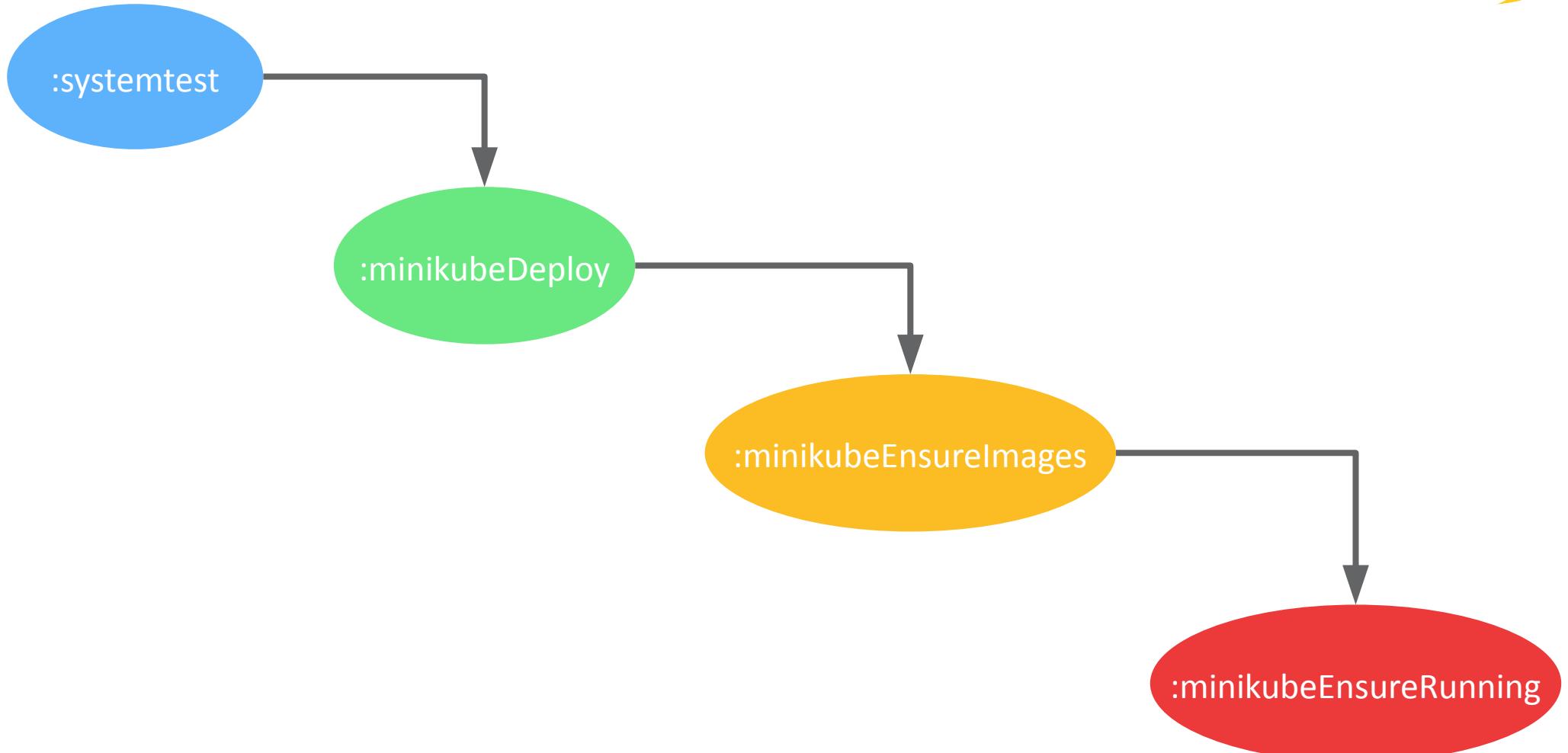
systemtest.sh

# Maven native solution

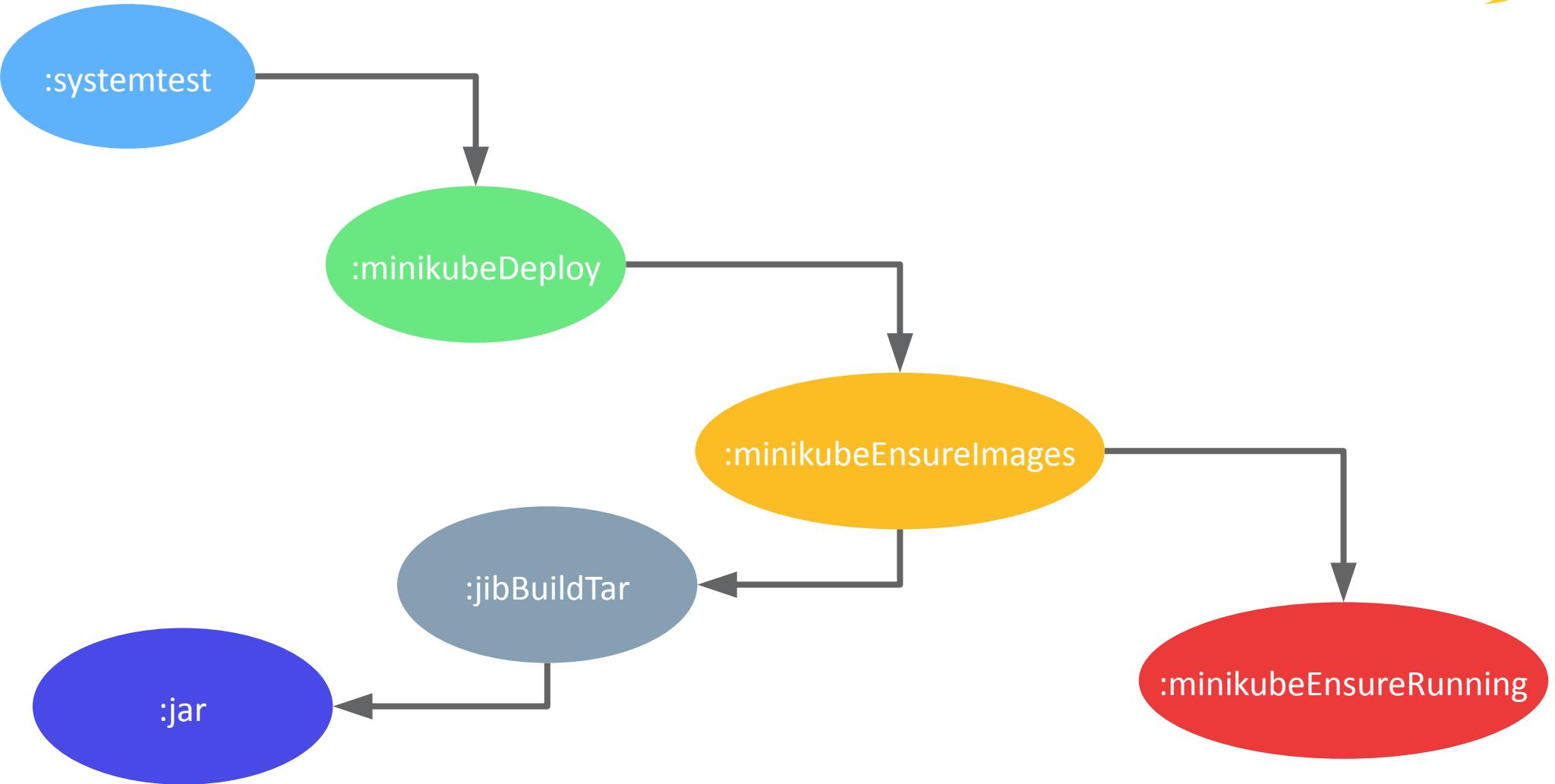
- 1) setup test cluster → pre-integration-test
- 2) compile the code → compile
- 3) build container image → package
- 4) transfer container image → pre-integration-test but after 1)
- 5) start container image → pre-integration-test but after 4)
- 6) run the test → integration-test

How could this look like with  
gradle?

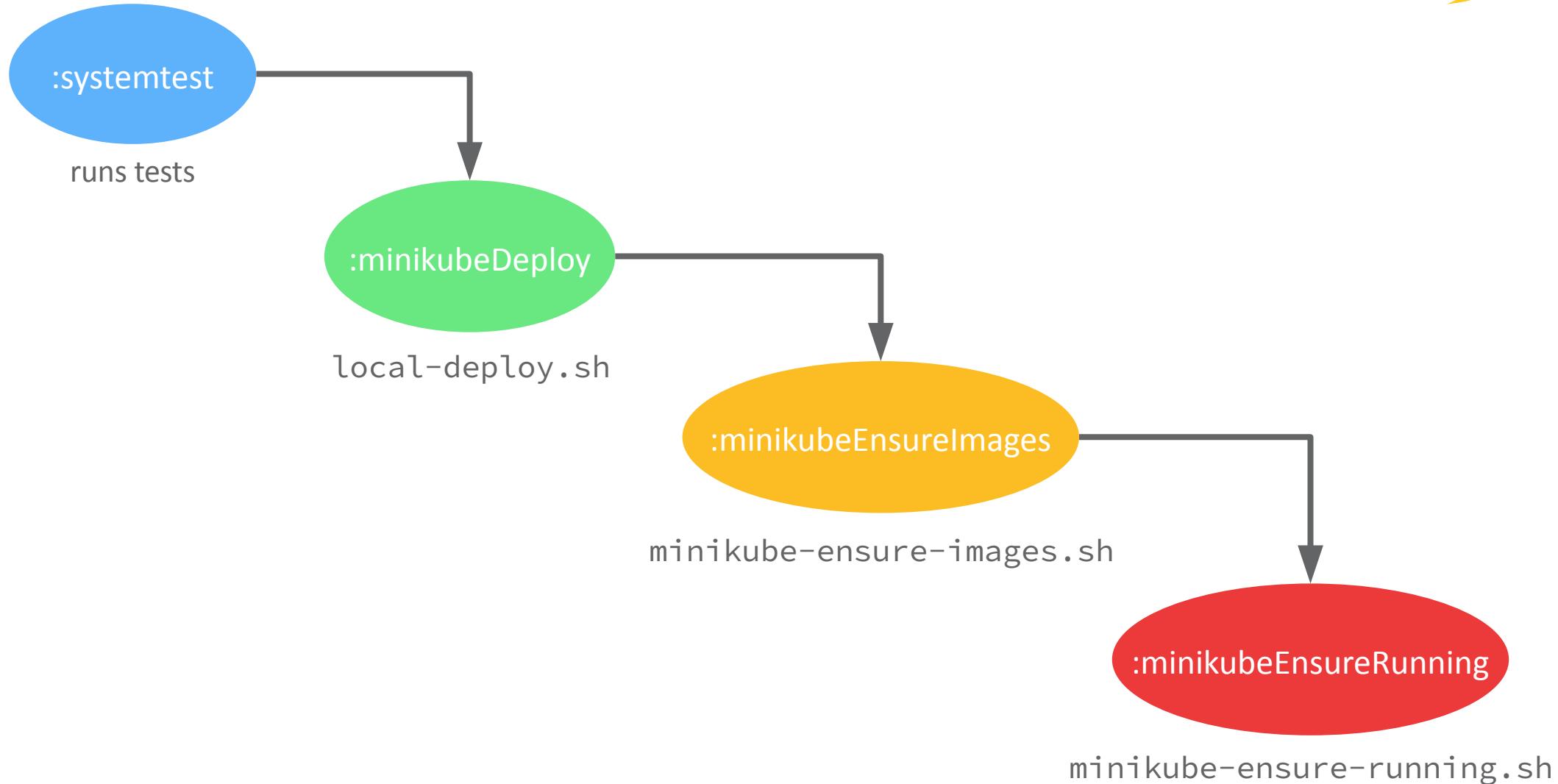
# Gradle Task Dependencies



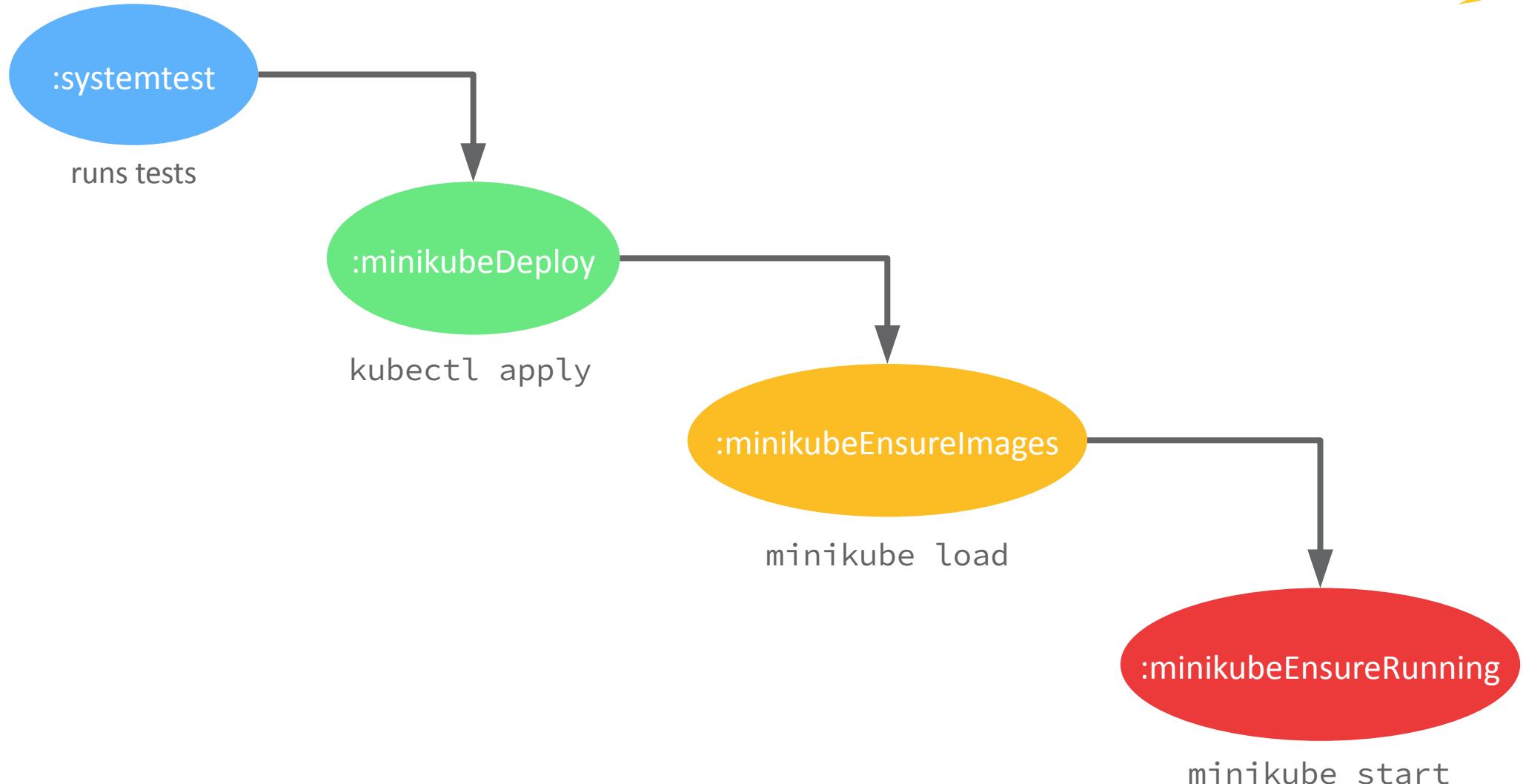
# Gradle Task Dependencies



# Gradle Task Content



# Gradle Task Content



# What have we achieved?

» minikube  
delete



more realism



Credit Sander Sammy on Unsplash



Credit Jaromír Kavan on Unsplash

# more realism

- ▶ fewer diffs local vs prod
  - more realistic tests
  - less room for errors
  - better understanding



# Quick Dev Cycle

- ▶ test very high integration
- ▶ test high reliability features like Auto Scalers, HA, ...
- ▶ use K8S API
- ▶ low pain when adding services



# Recommendations

- ▶ bring lots of RAM →  $\geq 32\text{GB}$  → 64GB recommended
- ▶ standard stack project → use skaffold
- ▶ have good network → use proper cluster per dev
- ▶ do DevOps
- ▶ maybe try K3S or KIND ?

# Credits & Links

- ▶ Example Project: <https://github.com/gitreelike/quickK8sFlow>
- ▶ Contact Me: immanuel . sims "ÄT" akquinet de



¿ QUESTIONS ?