akquinet
WIR. WISSEN. WIE.

Quick Response Kubernetes Dev Workflow

Immanuel Sims
BED-Con 2023, Berlin

Agenda

akquinet
WIR. WISSEN. WIE.

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

Motivation

akquinet
WIR. WISSEN. WIE.

Motivation

akquinet
WIR. WISSEN. WIE.

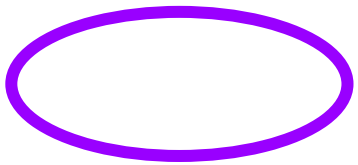
Keycloak

Motivation

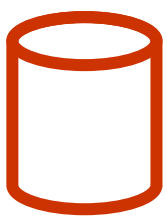
Motivation

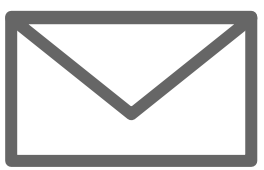
Motivation

akquinet
WIR. WISSEN. WIE.

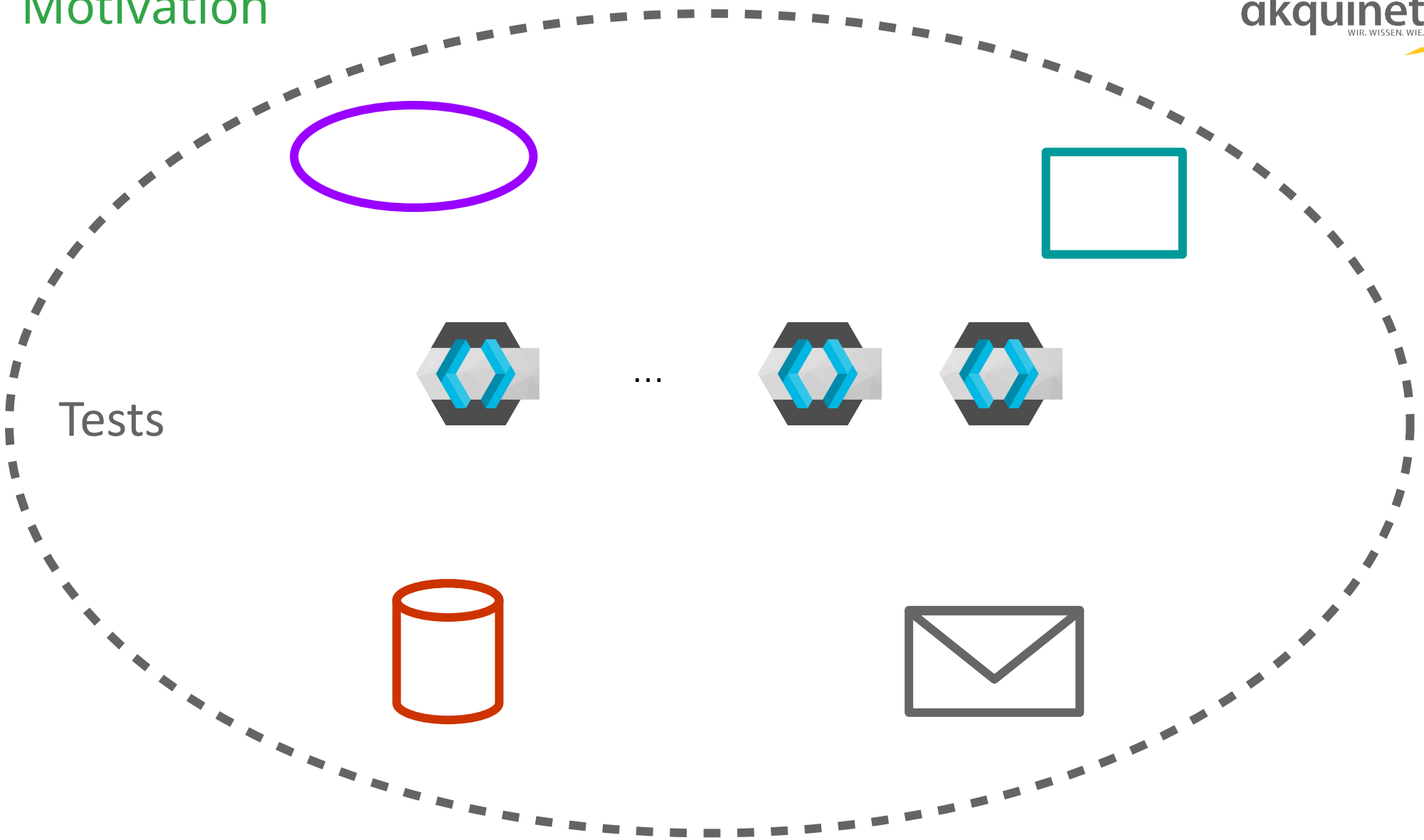


...

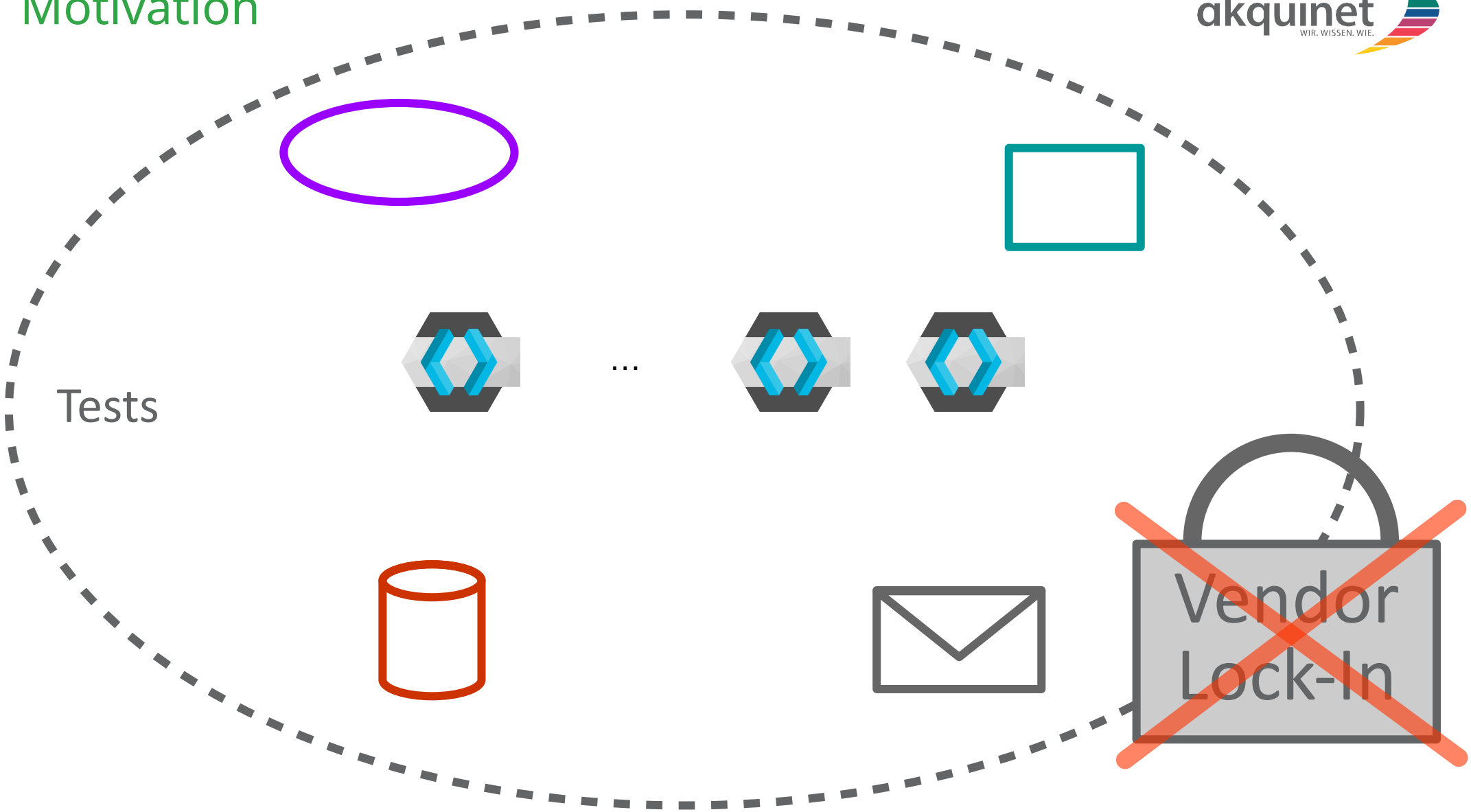




Motivation



Motivation



Requirements

akquinet
WIR. WISSEN. WIE.

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

Requirements

akquinet
WIR. WISSEN. WIE.

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

Requirements

akquinet
WIR. WISSEN. WIE.

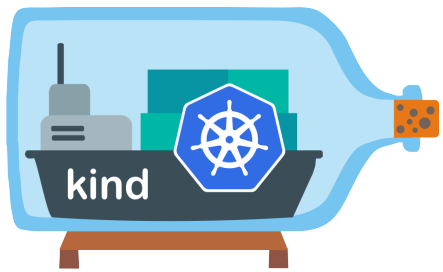
easy to install	→	container
easy to scale	→	container orchestration
no vendor lock in	→	K8S
quick & realistic E2E testing	→	K8S for each dev
want to use CRDs	→	definitely K8S for each dev

Tooling Options

akquinet
WIR. WISSEN. WIE.

minikube

SKAFFOLD



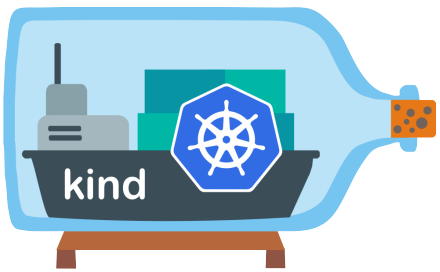
Google Cloud Platform

Tooling Options

akquinet
WIR. WISSEN. WIE.

minikube

SKAFFOLD

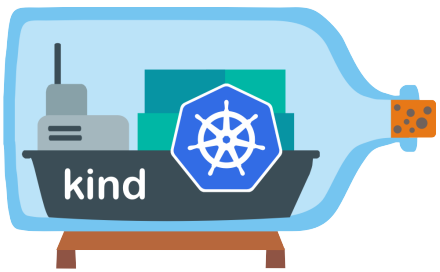


Google Cloud Platform

Tooling Options

akquinet
WIR. WISSEN. WIE.

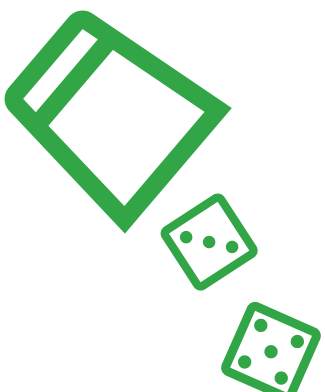
minikube

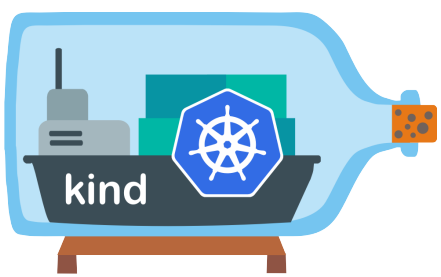


Google Cloud Platform

Tooling Options

minikube



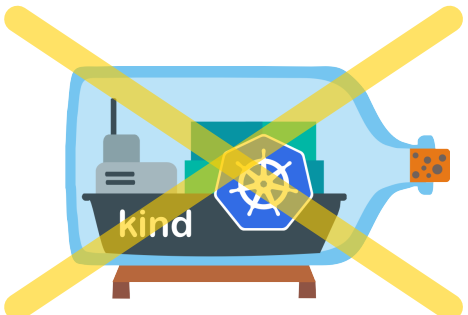


Credits for Cup and Dice: Google Fonts

Tooling Options

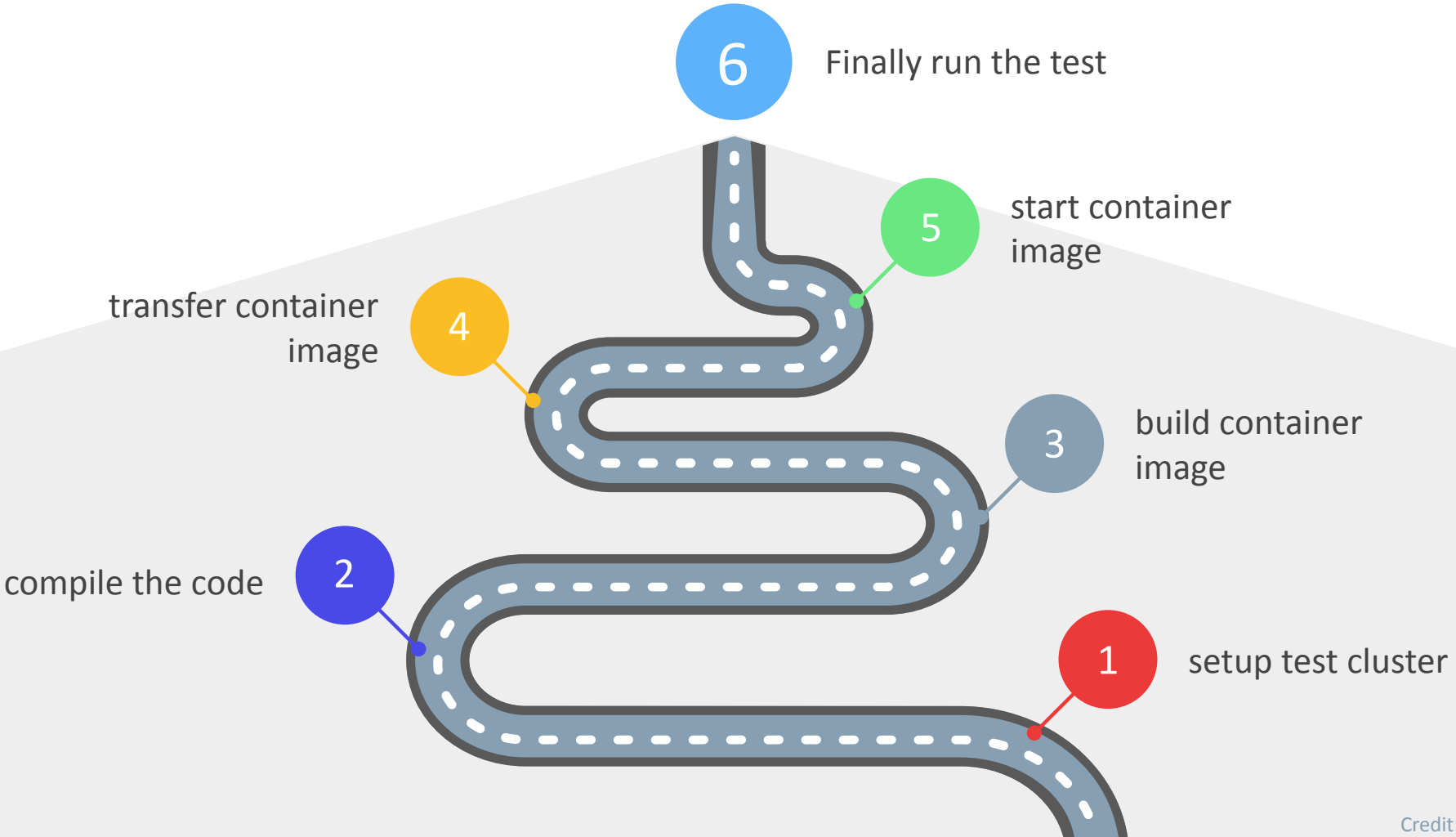
- driver options
- configurable k8s version
- persistent storage
- ...

minikube



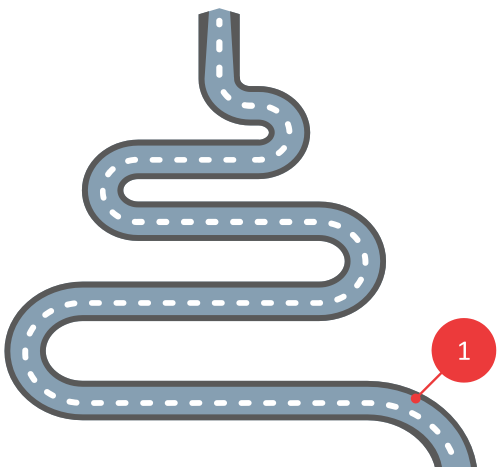
What needs to be done?

Running a System Test



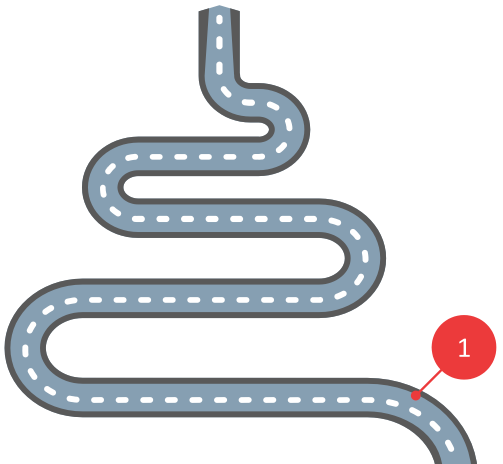
1) set up access to test cluster

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



1) set up access to test cluster

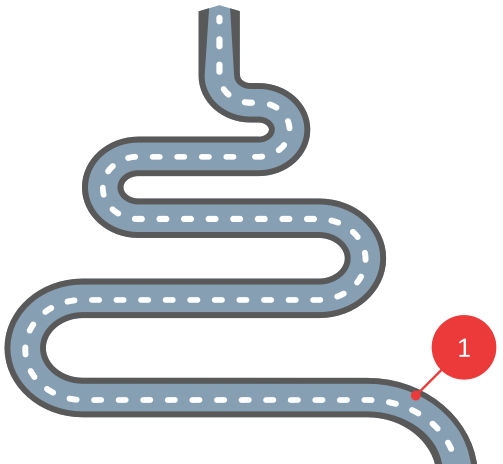
- `» minikube start`
- options for
  - `--memory`
  - `--cpus`
  - `--container-runtime`
  - `--kubernetes-version`



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

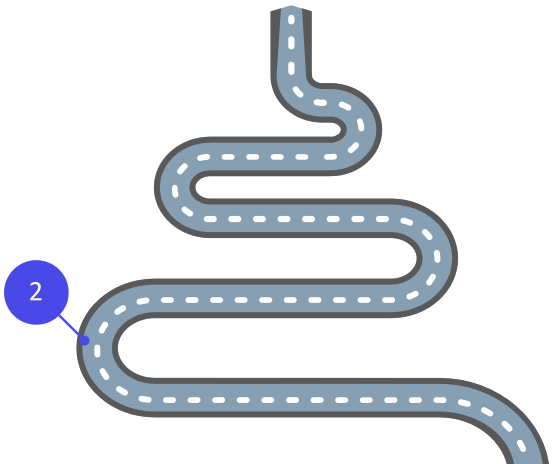
All Happy:



2) compile code

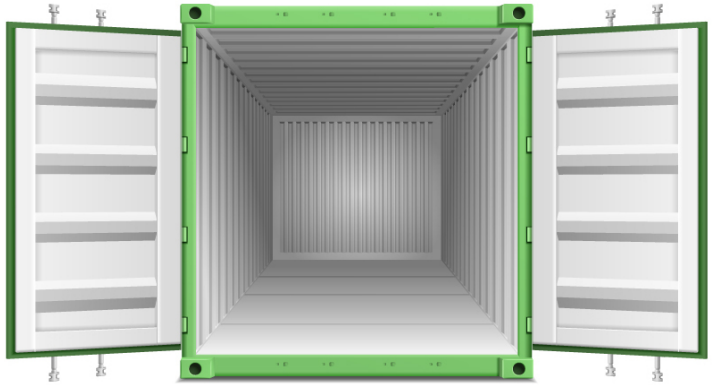
- business as usual

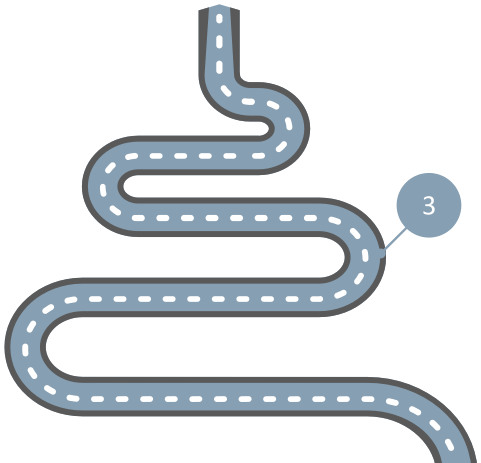
Credit: Imageflip



3) build container image

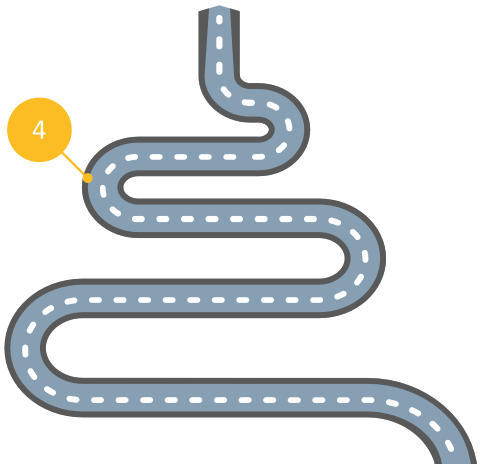
- Options:
  a) » `docker build`
  b) jib





Credit: macrovector on Freepik

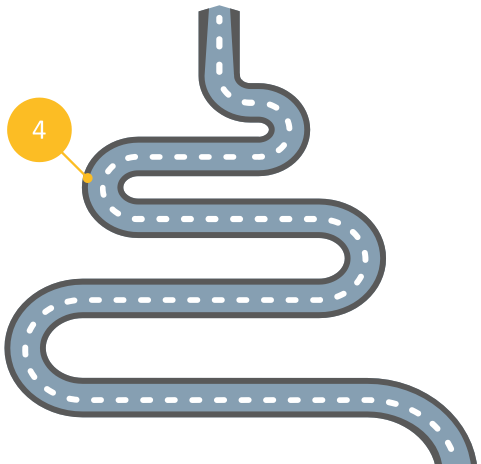
4) transfer container image



Credit: vecstock on Freepik

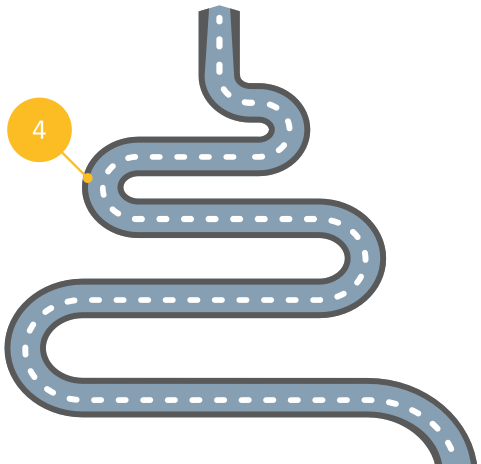
4) transfer container image

- Options:
  a) » `docker push`



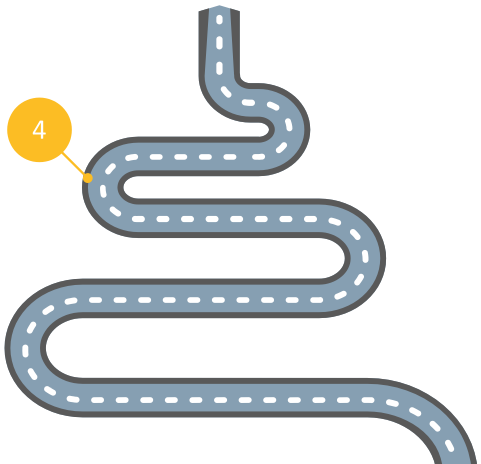
4) transfer container image

- 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

akquinet
WIR. WISSEN. WIE.

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

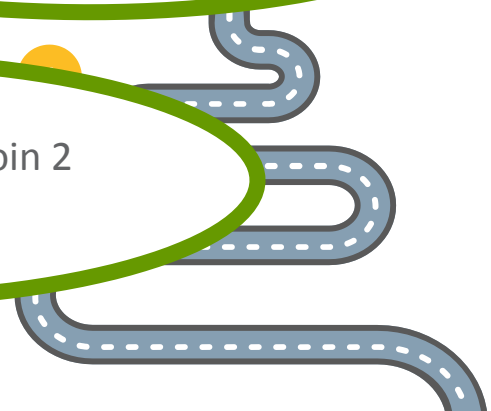
- Options:
  - a) ~~» `docker push`~~
  - b) » `minikube load`
  - c) » `minikube docker-env` with
    - → » `docker build`
    - → » `jib:dockerBuild`

bad because rare bugs

good because it's flexible

can't do cache optimizations

good because it can join 2 steps

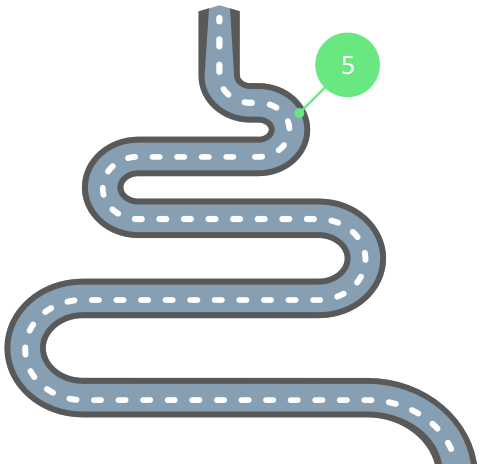


5) start container image

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

Kustomize.io

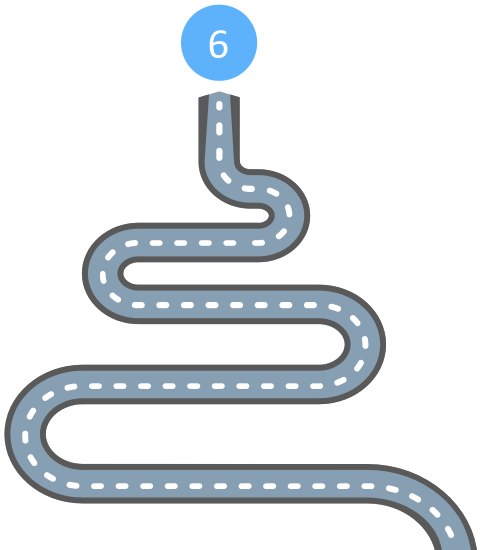
HELM



6) actually run systemtest

- business as usual

Credit: Imageflip



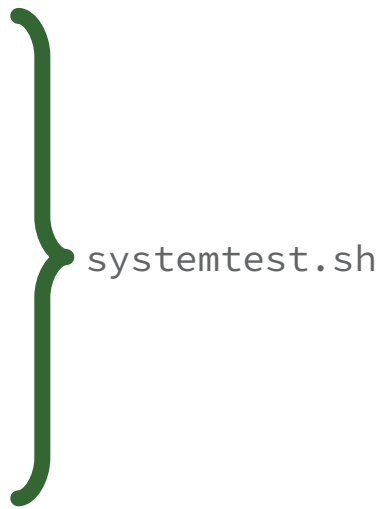
akquinet
WIR. WISSEN. WIE.

How could this look like with

maven ?

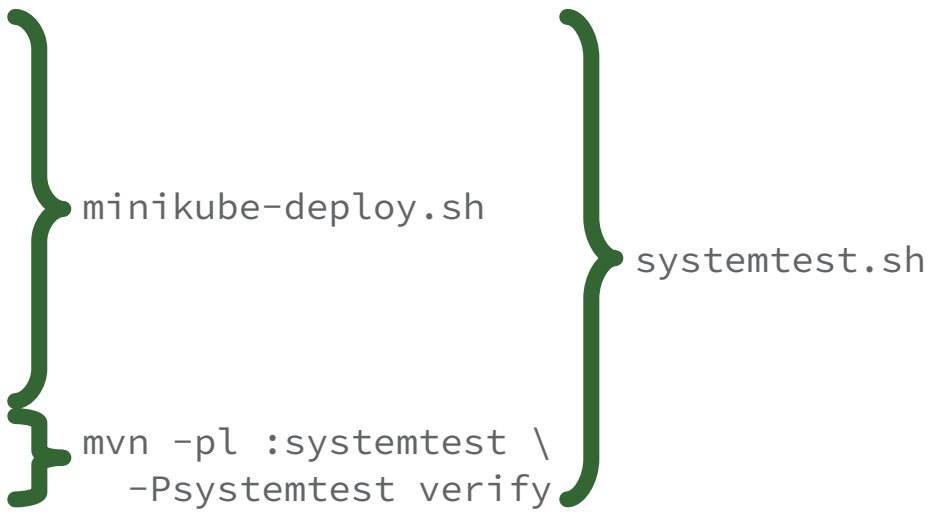
Maven

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



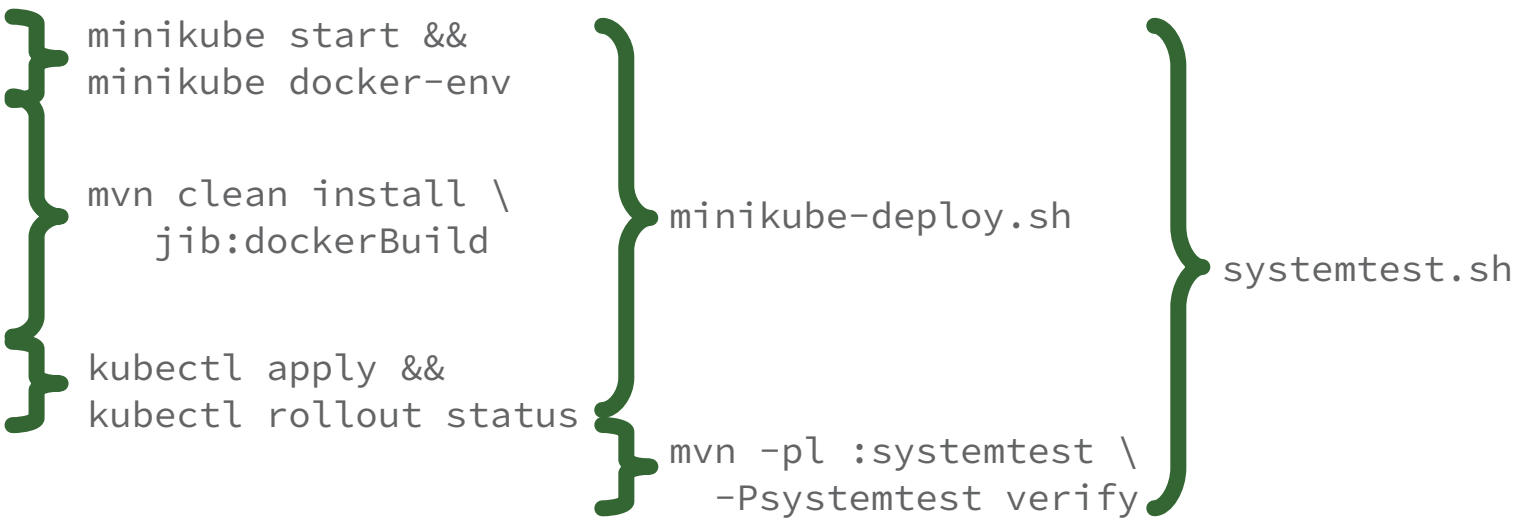
Maven

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



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



Maven native solution

akquinet
WIR. WISSEN. WIE.

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

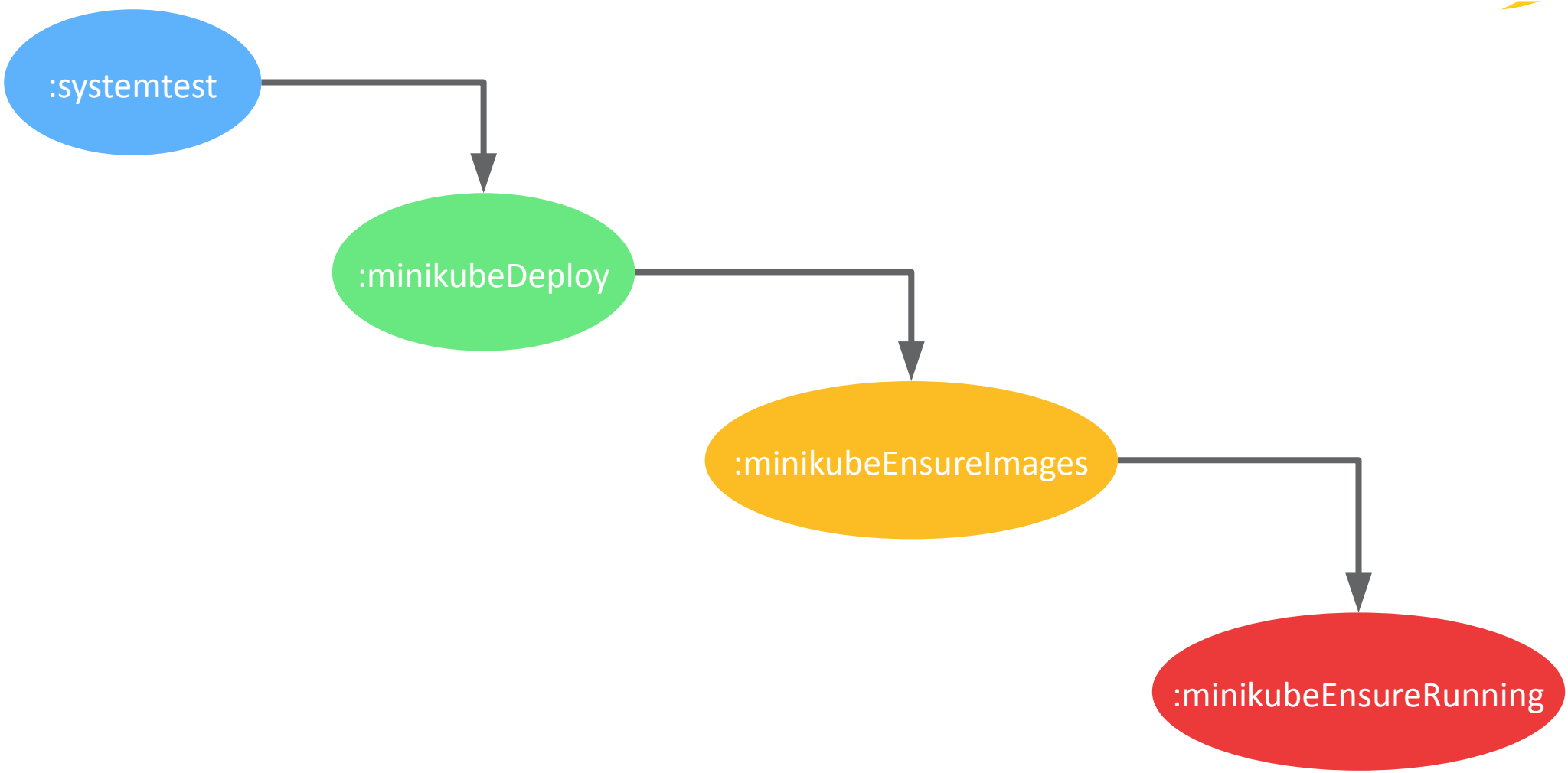
akquinet
WIR. WISSEN. WIE.

How could this look like with

gradle?

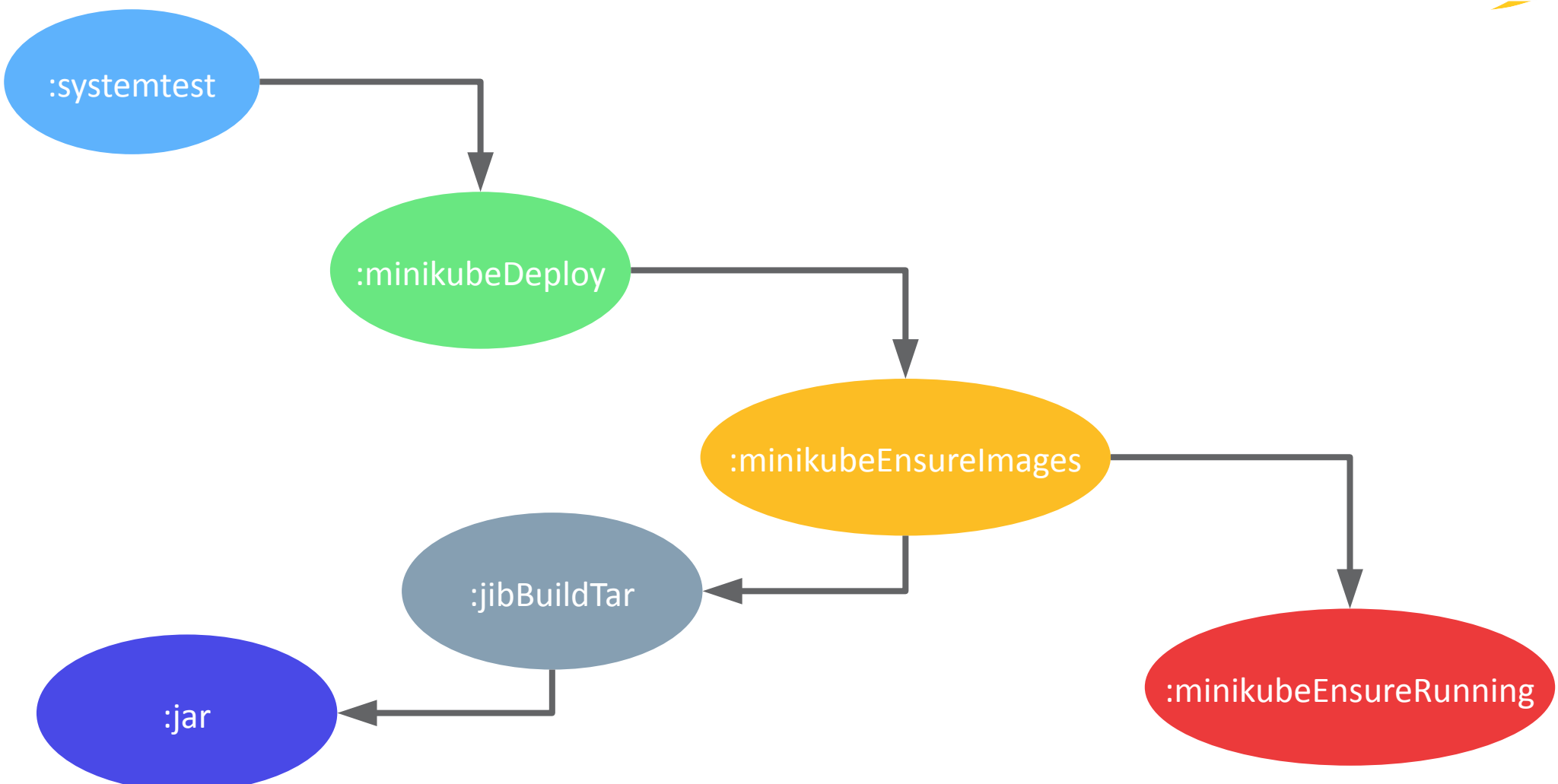
Gradle Task Dependencies

akquinet
WIR. WISSEN. WIE.

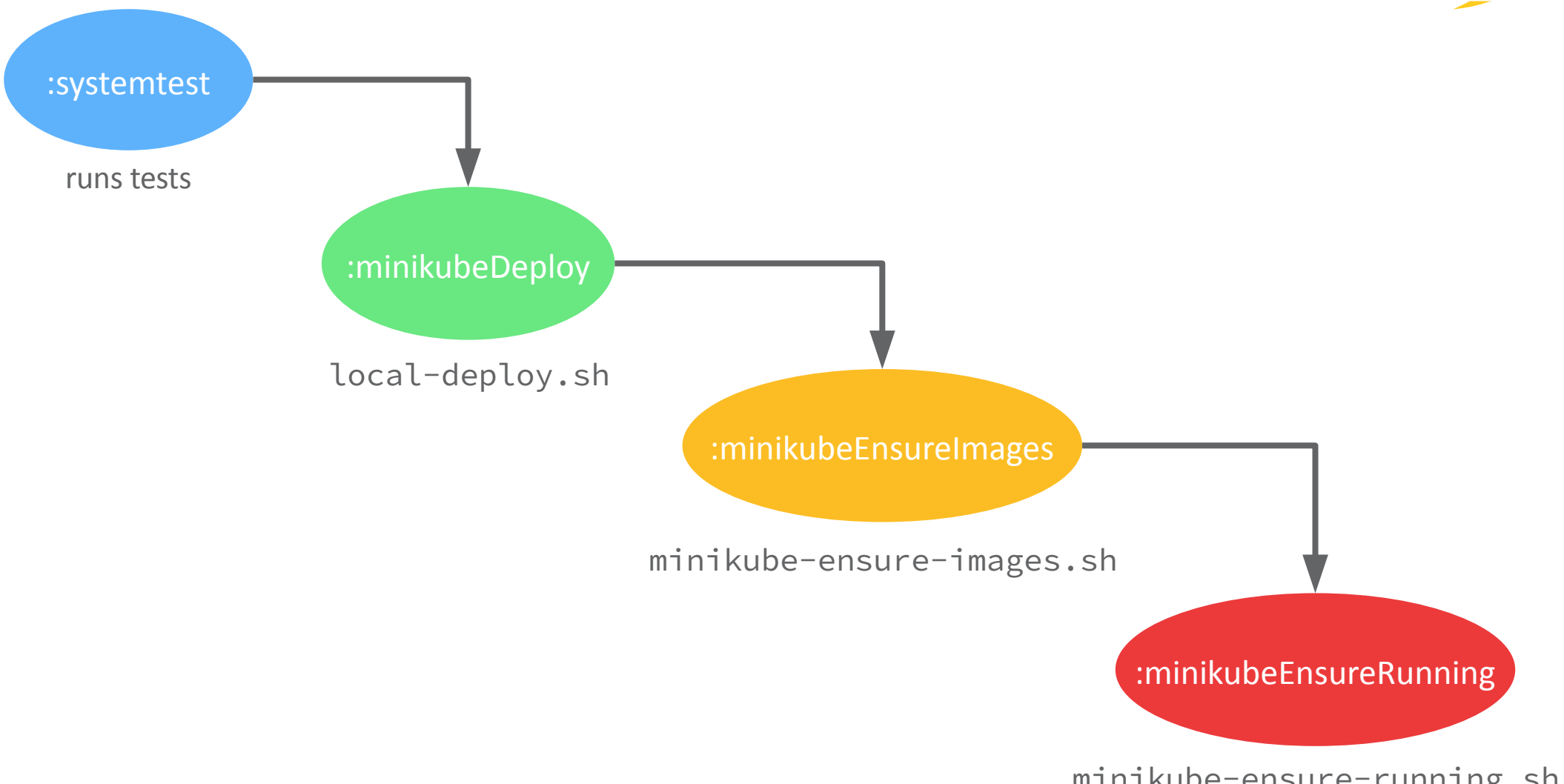


Gradle Task Dependencies

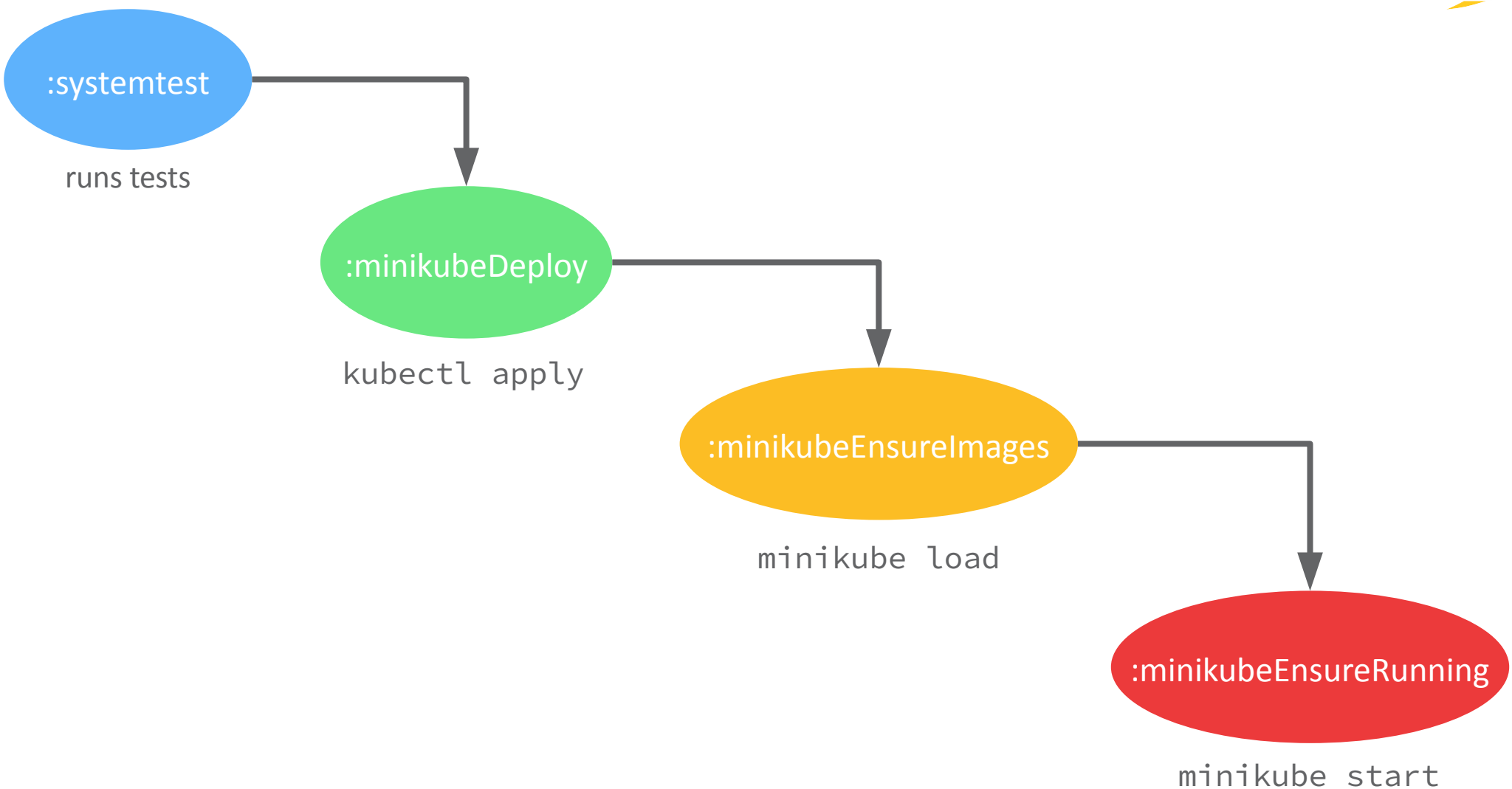
akquinet
WIR. WISSEN. WIE.



Gradle Task Content

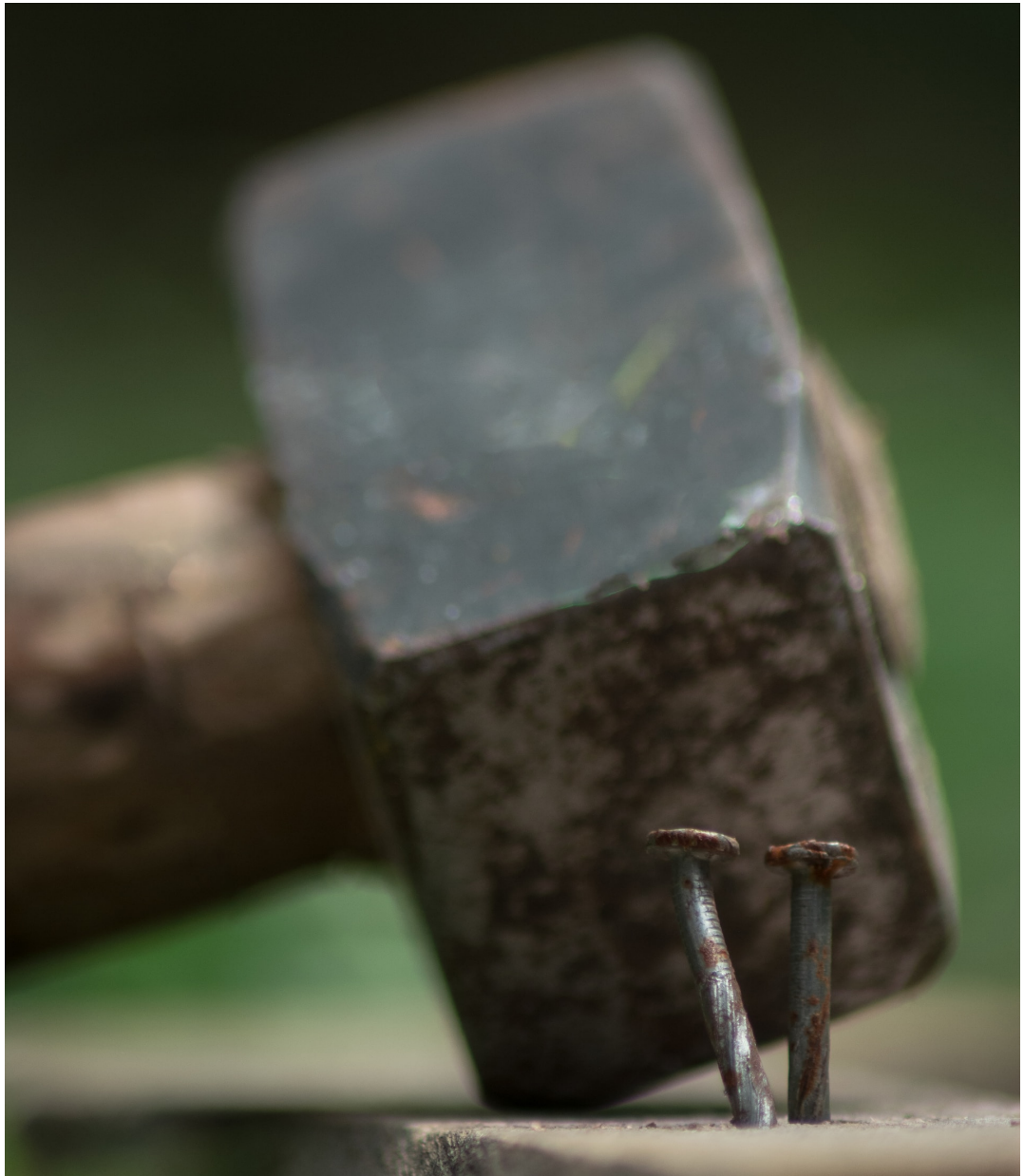


Gradle Task Content



What have we achieved?

» minikube
delete



akquinet
WIR. WISSEN. WIE.

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

Credit: John Cameron on Unsplash

Recommendations

akquinet
WIR. WISSEN. WIE.

- bring lots of RAM → >=32GB → 64GB recommended
- standard stack project → use skaffold
- have good network → use proper cluster per dev
- do both dev & ops
- maybe try KIND ?

Credits & Links

- Example Project: https://github.com/gitreelike/quickK8sFlow
- Contact Me: immanuel . sims ˝ÄT˝ akquinet de

akquinet
WIR. WISSEN. WIE.

¿ QUESTIONS ?