Microservices Lesson 3

Methods and Techniques for Software Development

2019/2020

P.PORTO

Microserviços Aula 3

Methods and Techniques for Software Development

2019/2020

THE LIFE OF A SOFTWARE ENGINEER.

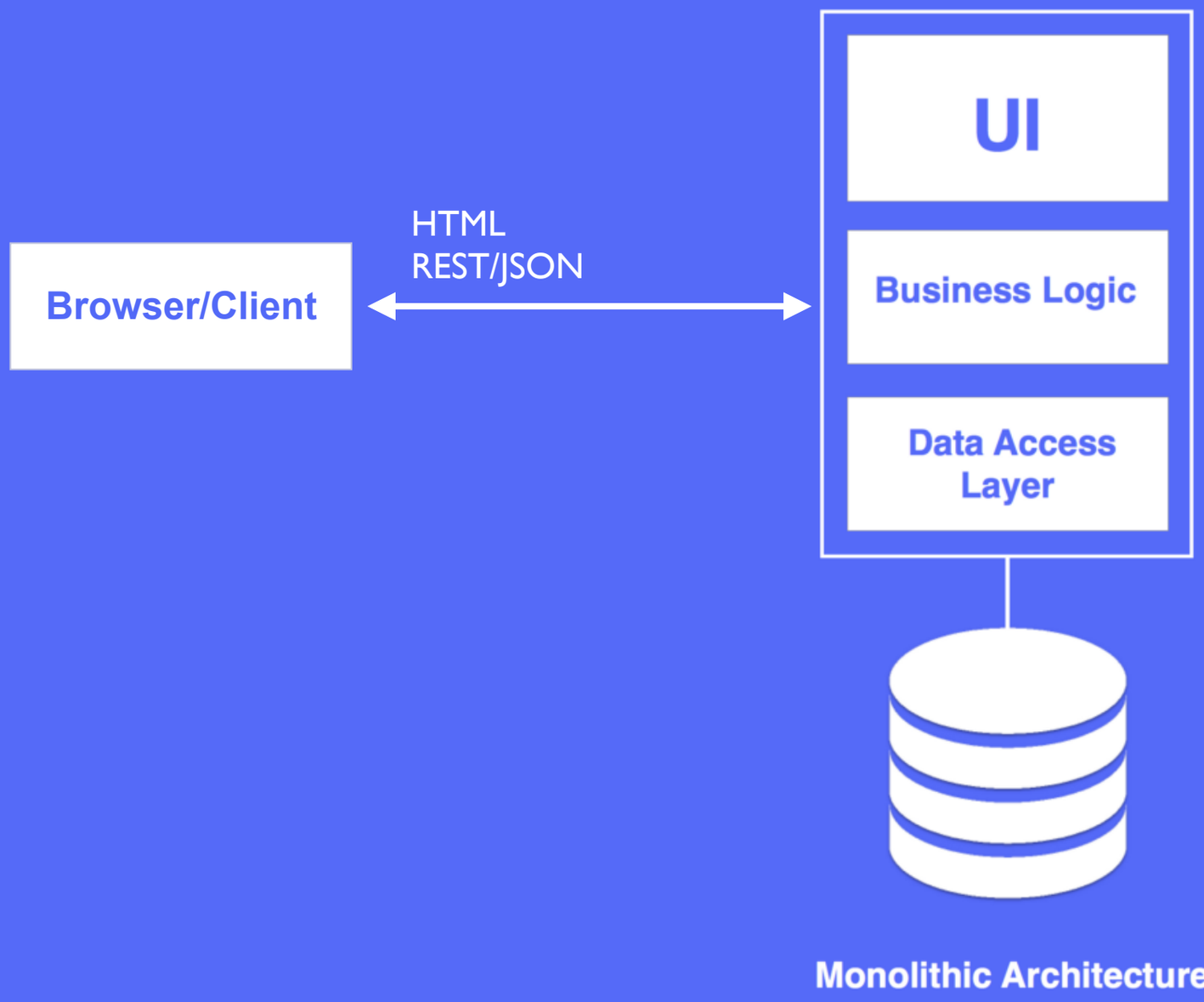
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.

MUCH LATER...

OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?

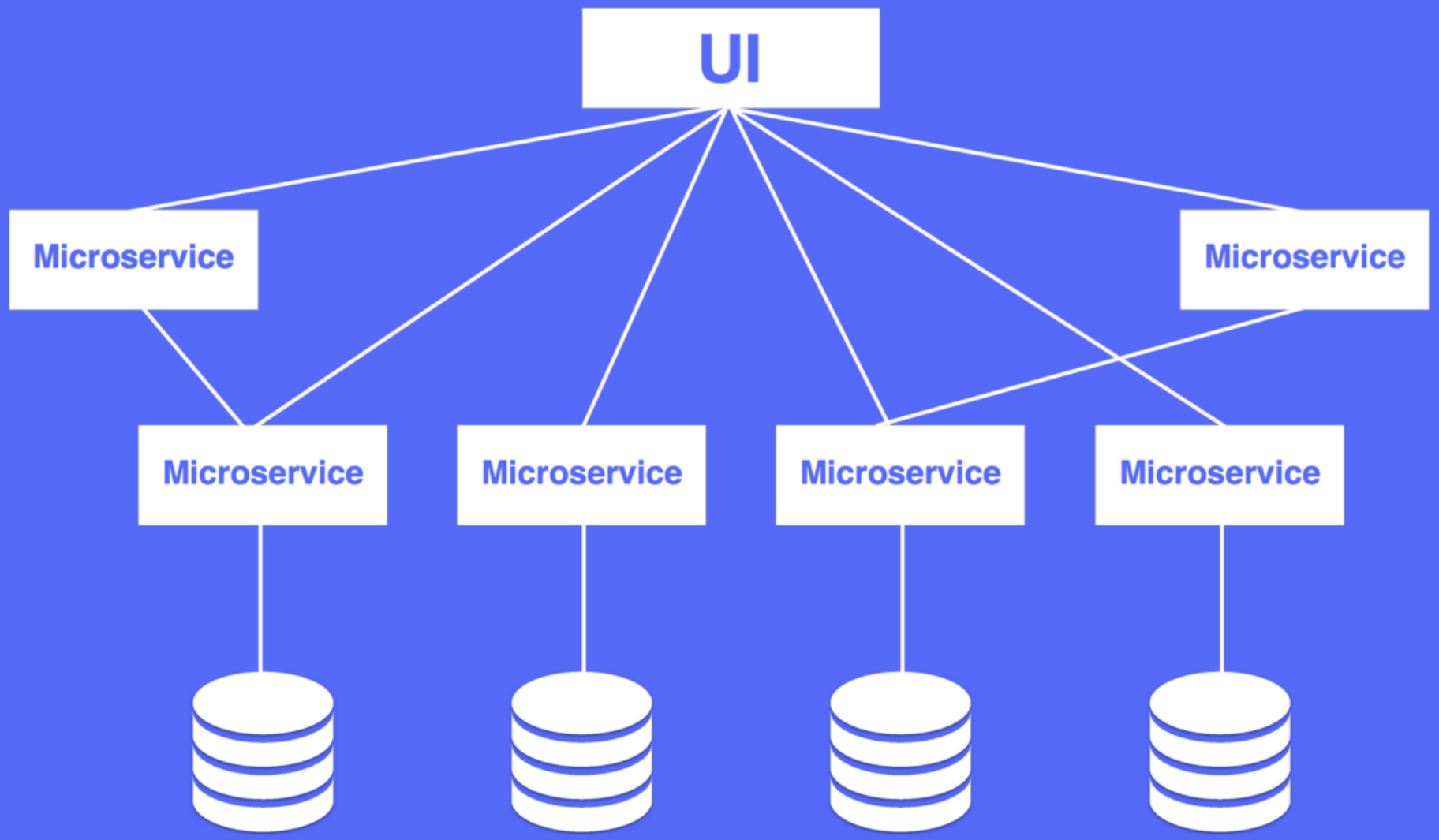
MANU


WHAT?



Monolithic Architecture

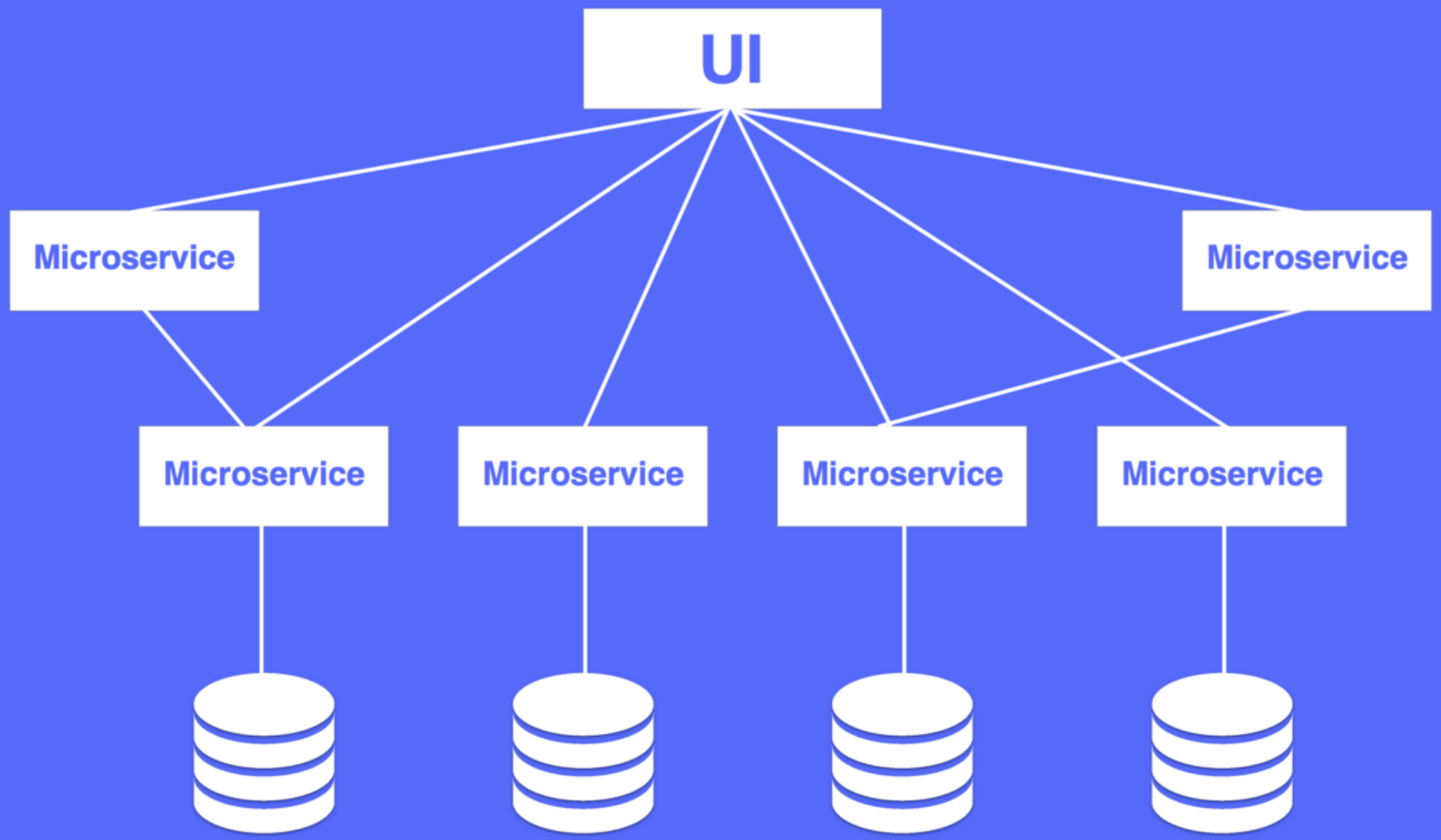

H2Only



Microservices Architecture


PETRONAS
Santander
Claro
5
44

Monolithic Architecture



Microservices Architecture

P.PORTO

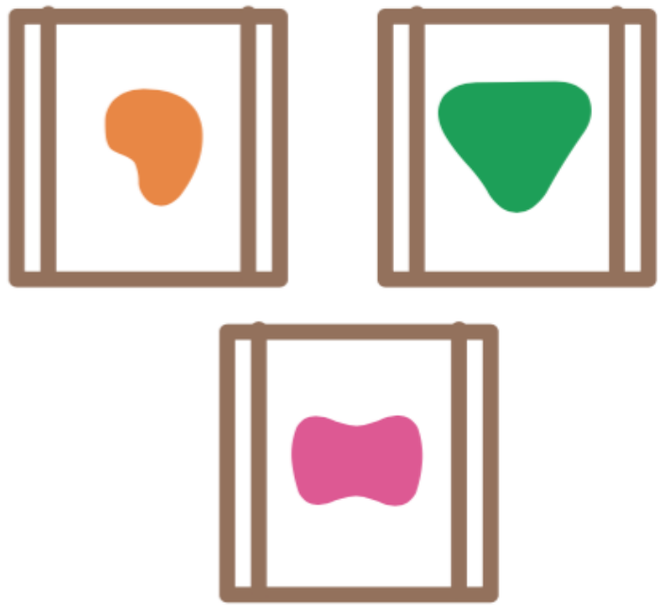
ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

The microservice architecture is an ***architectural style*** *that* ***structures*** *an application as a* ***set of loosely coupled, services*** *organized around* ***business capabilities***

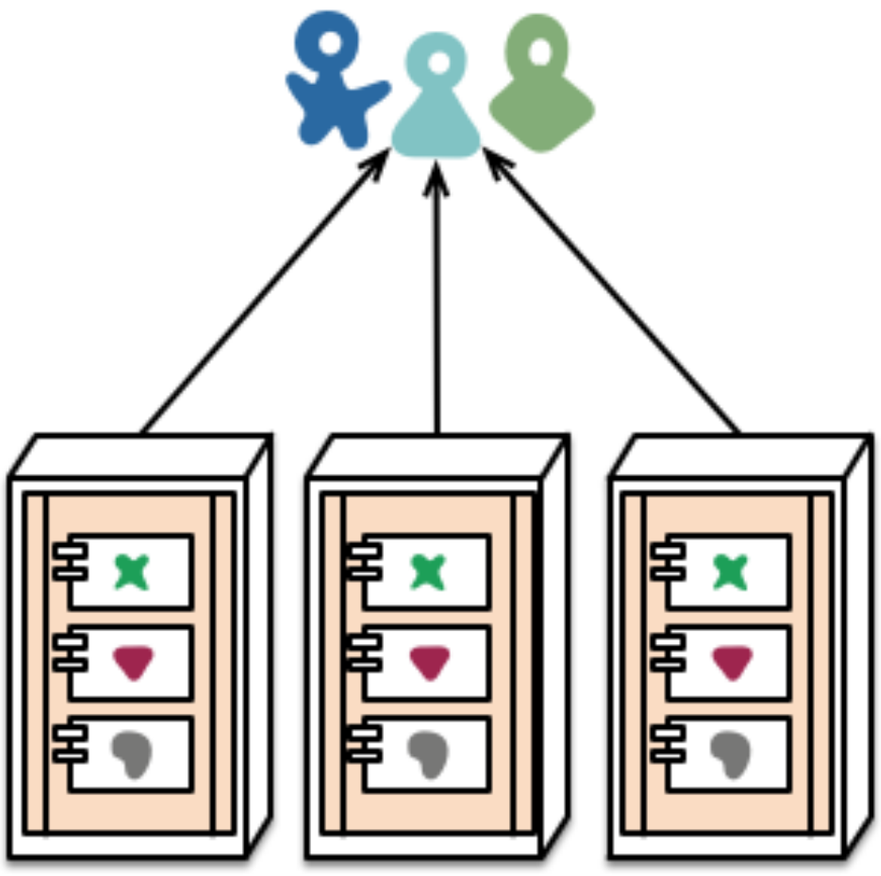
Simple and Lightweight

A monolithic application puts all its functionality into a single process...

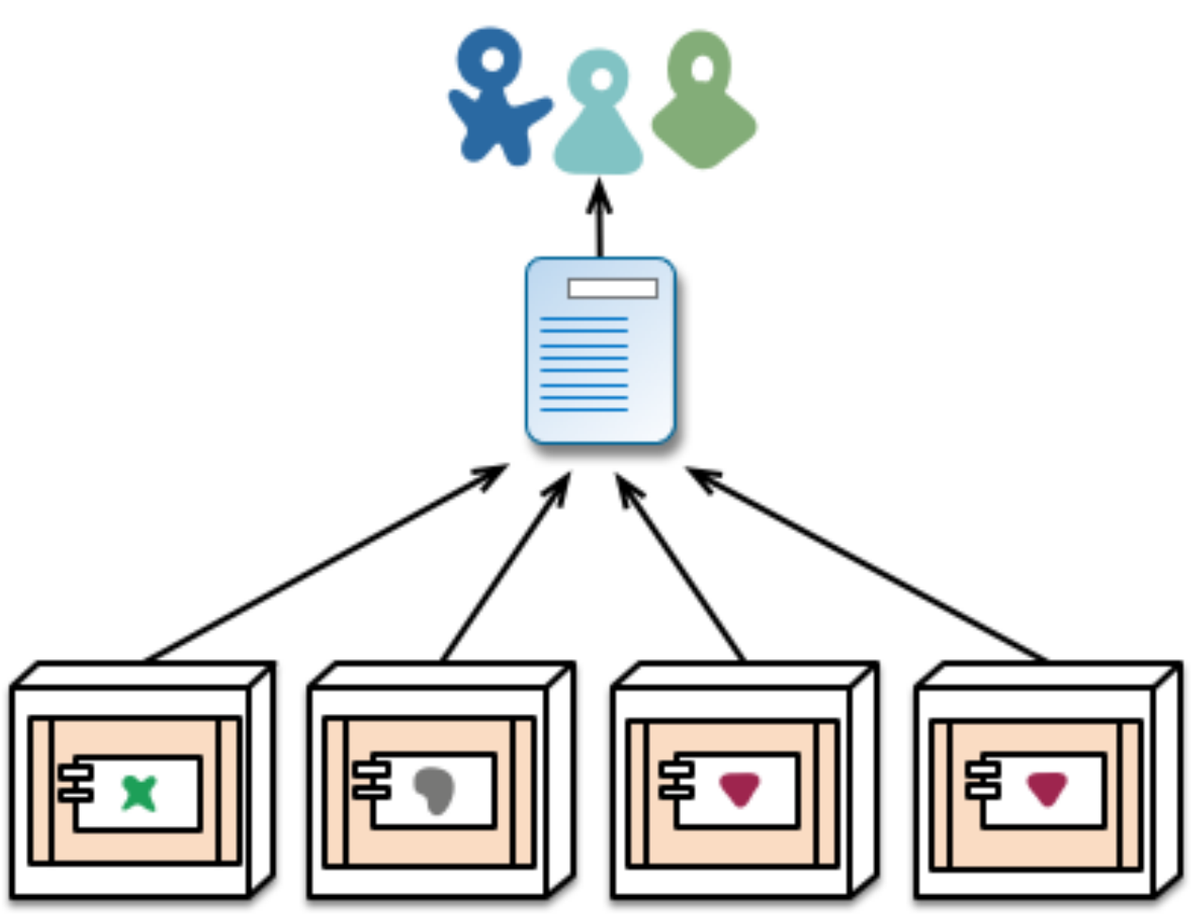
A microservices architecture puts each element of functionality into a separate service...



Independent Processes



monolith - multiple modules in the same process



microservices - modules running in different processes

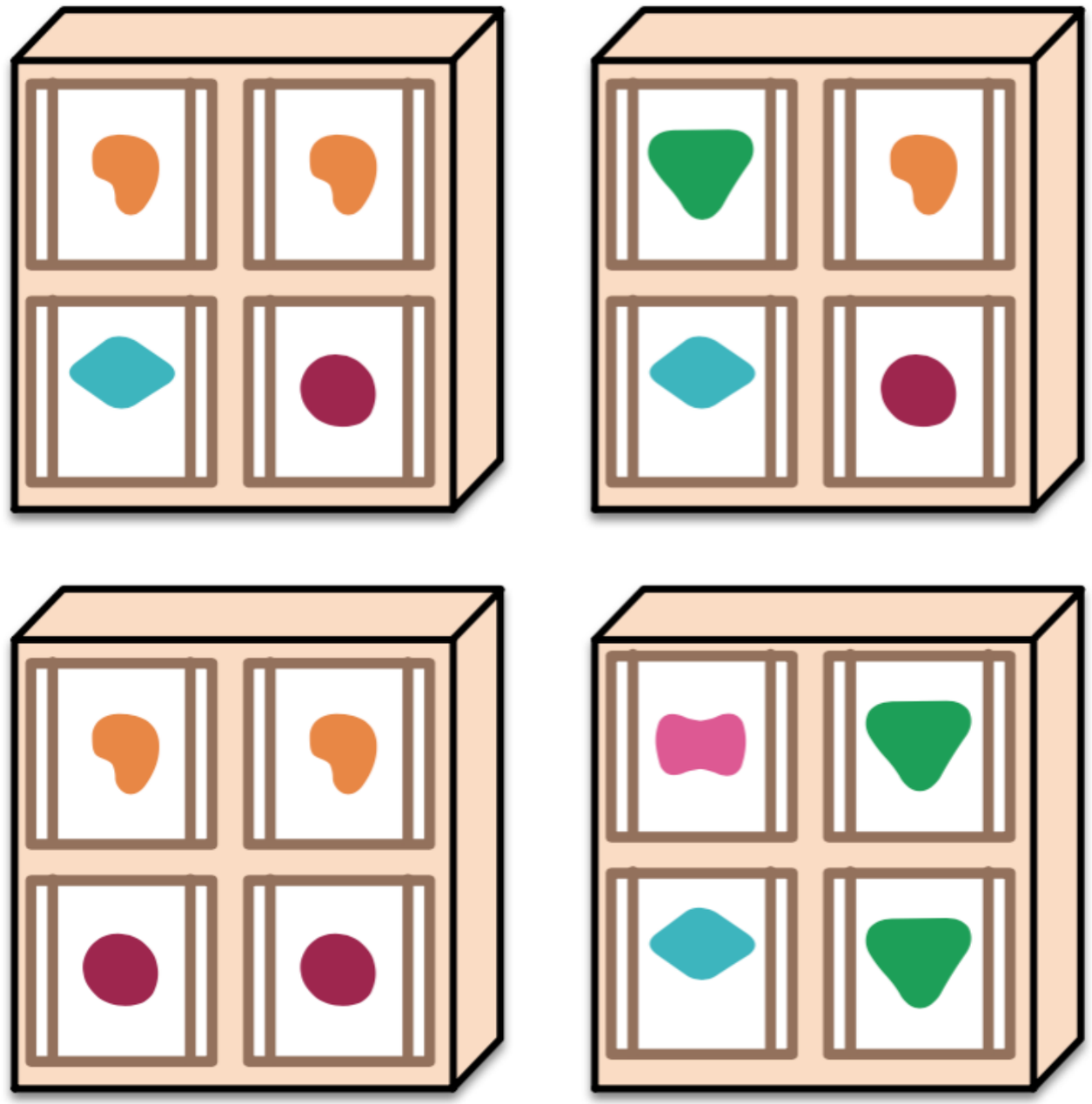
P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Language Agnostic

“be of the web”

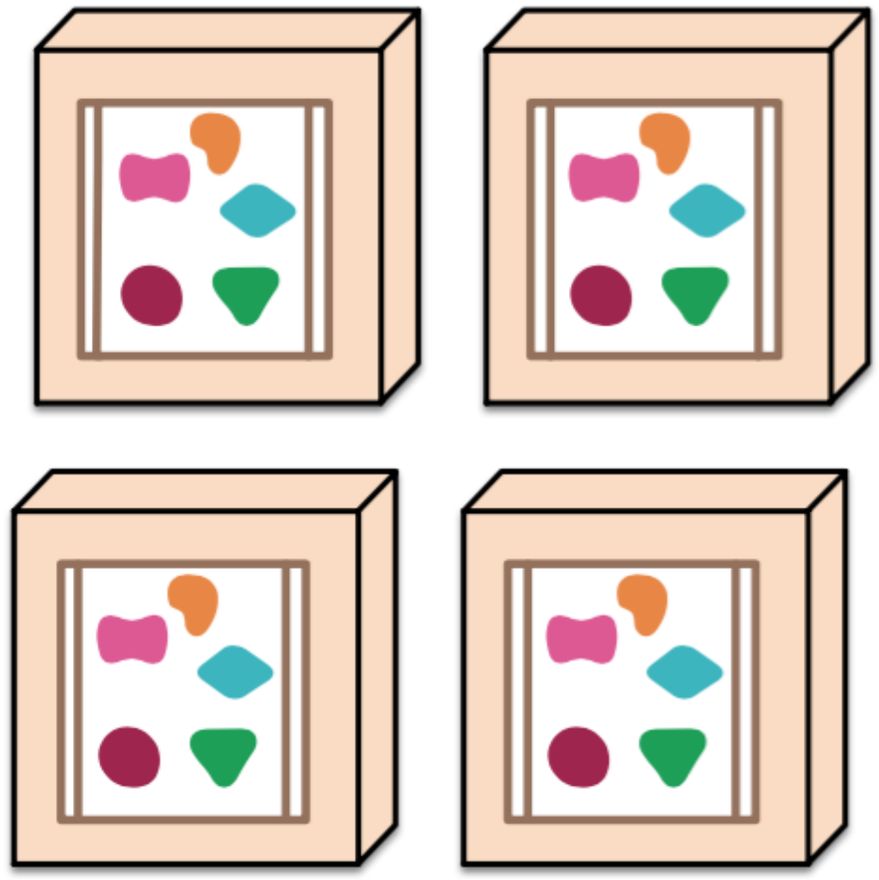
Decoupled



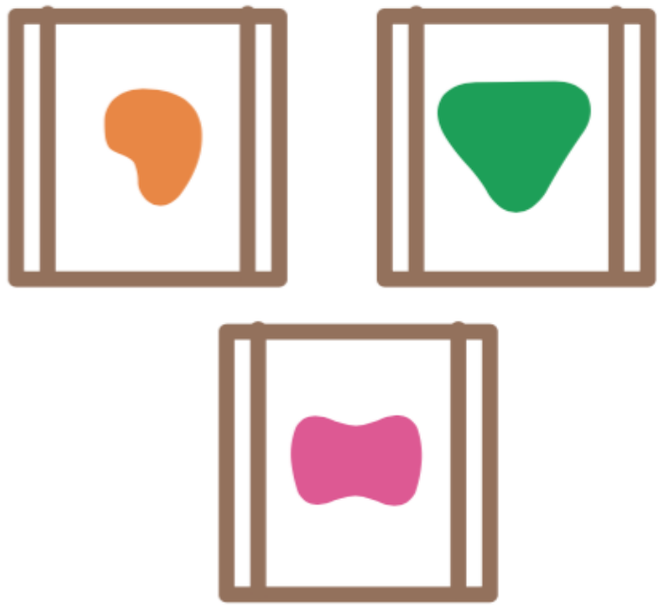
Scaling

A monolithic application puts all its functionality into a single process...

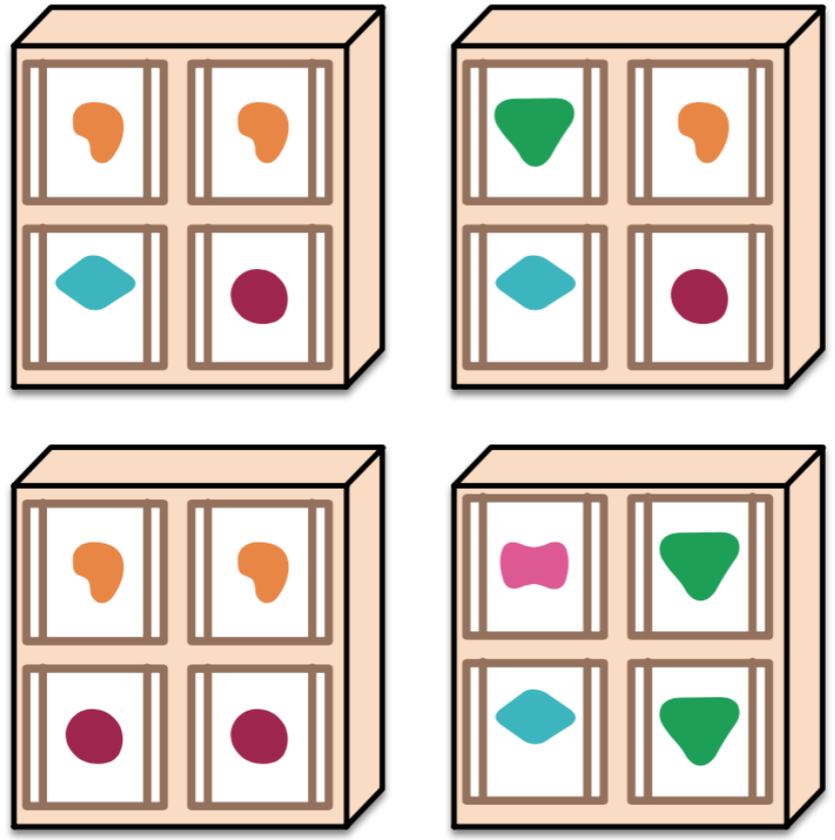
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



Two levels of architecture

Application-level

Services

Inter-service glue: interfaces and communication mechanisms

Slow changing

Service-level

Internal architecture of each service

Each service could use a different technology stack

Pick the best tool for the job

Rapidly evolving

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Microservices enable continuous delivery/deployment

Process: *Continuous delivery/ deployment*

Enables

Enables

services improve testability and deployability

Successful Software Development

Organization:
Small, agile, autonomous, cross functional teams

Enables

Architecture:
Microservice architecture

Teams own services

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Evolve the technology stack

- Pick a new technology when
  - Writing a new service
  - Making major changes to an existing service
- Let's you experiment and fail safely

Drawbacks of micro services

- Development:
  - Interprocess communication - IPC
  - Partial failure
    - distributed data
- Testing
- Deployment
- …

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

- How to decompose an application into services?
- How to deploy an application's services?
- How to handle cross cutting concerns?
- Which communication mechanisms to use?
- How do external clients communicate with the services?
- How does a client discover the network location of a service instance?

- How to prevent a network or service failure from cascading to other services?
- How to maintain data consistency and implement queries?
- How to understand the behavior of an application and troubleshoot problems?
- How to make testing easier?
- How to implement a UI screen or page that displays data from multiple services?

kubernetes

Manage applications, not machines

- Open source, open API container **orchestrator**
- Supports **multiple cloud** and **bare-metal** environments
- Inspired and informed by Google's experiences and internal systems

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

What we need is a system to handle that complexity for us

without locking us into any one vendor or way of doing things

Cattle vs Pets

- Has a name
- Is unique or rare
- Personal attention
- If it gets ill, you make it better

- Has a number
- One is much like any other
- Run as a group
- If it gets ill, you make hamburgers

kubernetes

Desired State

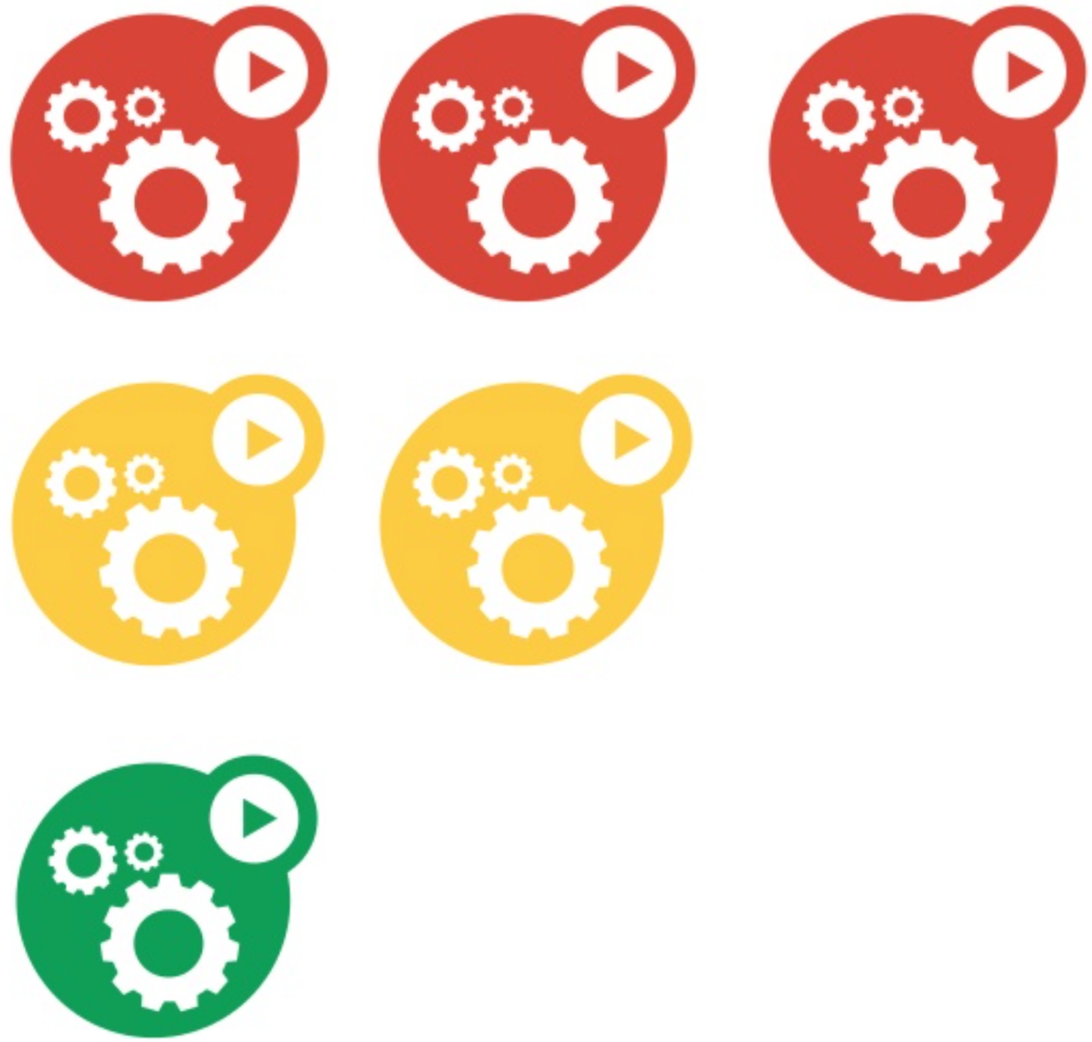
One of the core concepts of Kubernetes

Tell kubernetes what you want, not what to do!

P.PORTO

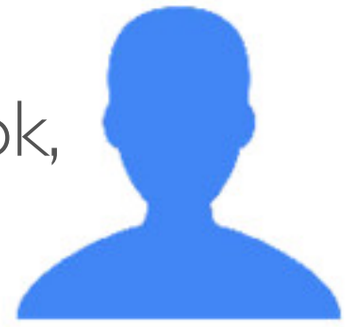
ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Desired States



```
$./create_docker_images.sh
$./lauch_frontend.sh  x 3
$./launch_services.sh  x 2
$./lauch_backend.sh  x 1
```

- **Under and imperative system**, you have a series of tasks
  - Create images
  - Create 3 frontends
  - Create 2 services, etc
- If something dies, something has to react
  - Under your worst case it's and admin that has to react (ok, maybe you have something automated)



P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Under a **desired state** you just say
I want 3 2 1

Then if something blows up, you're no longer in the desired state, so kubernetes will fix it!

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Employees, not Children

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Children vs Employees

- Go upstairs
- Get undressed
- Put on pijamas
- Brush your teeth
- Pick out 2 stories

- go get some sleep

kubernetes

External Services

Services

Rolling Update

Pods

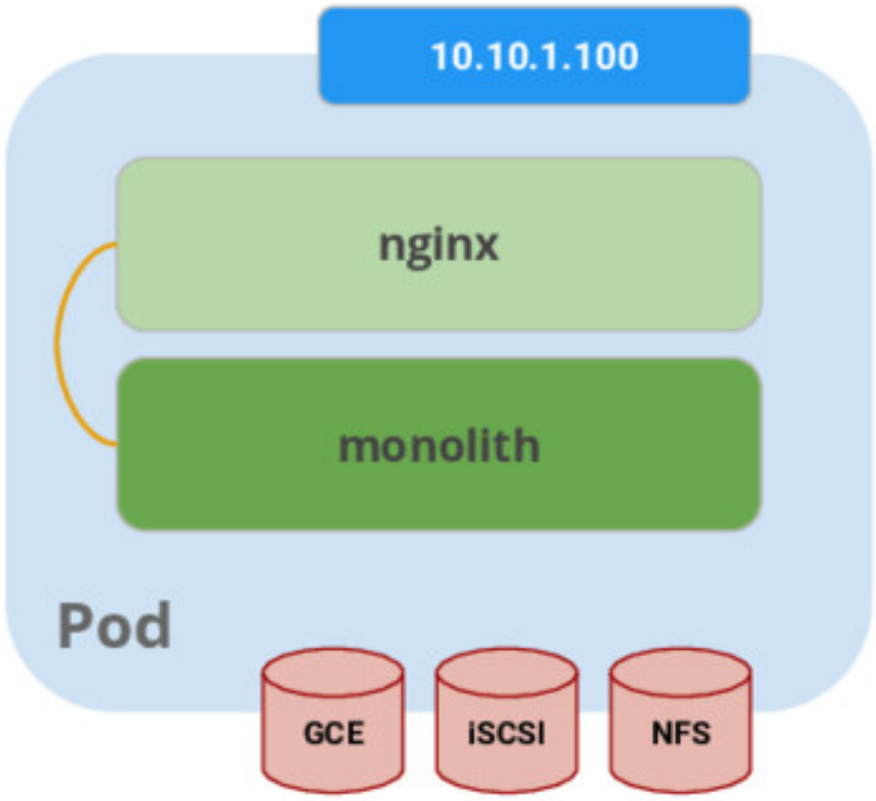
ReplicaSets

Labels & Selectors

Monitor & Health Checks

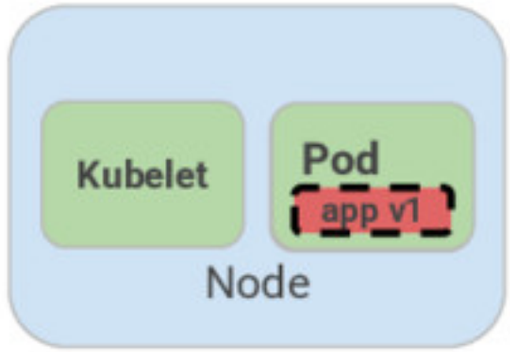
Pods

- **Small group** of containers & volumes
- **Tightly** coupled
- The atom of scheduling & placement
- Shared namespace
  - share IP address & localhost
  - share IPC, etc
- Managed lifecycle
  - bound to a node, restart in place
  - can die, cannot reborn with same ID



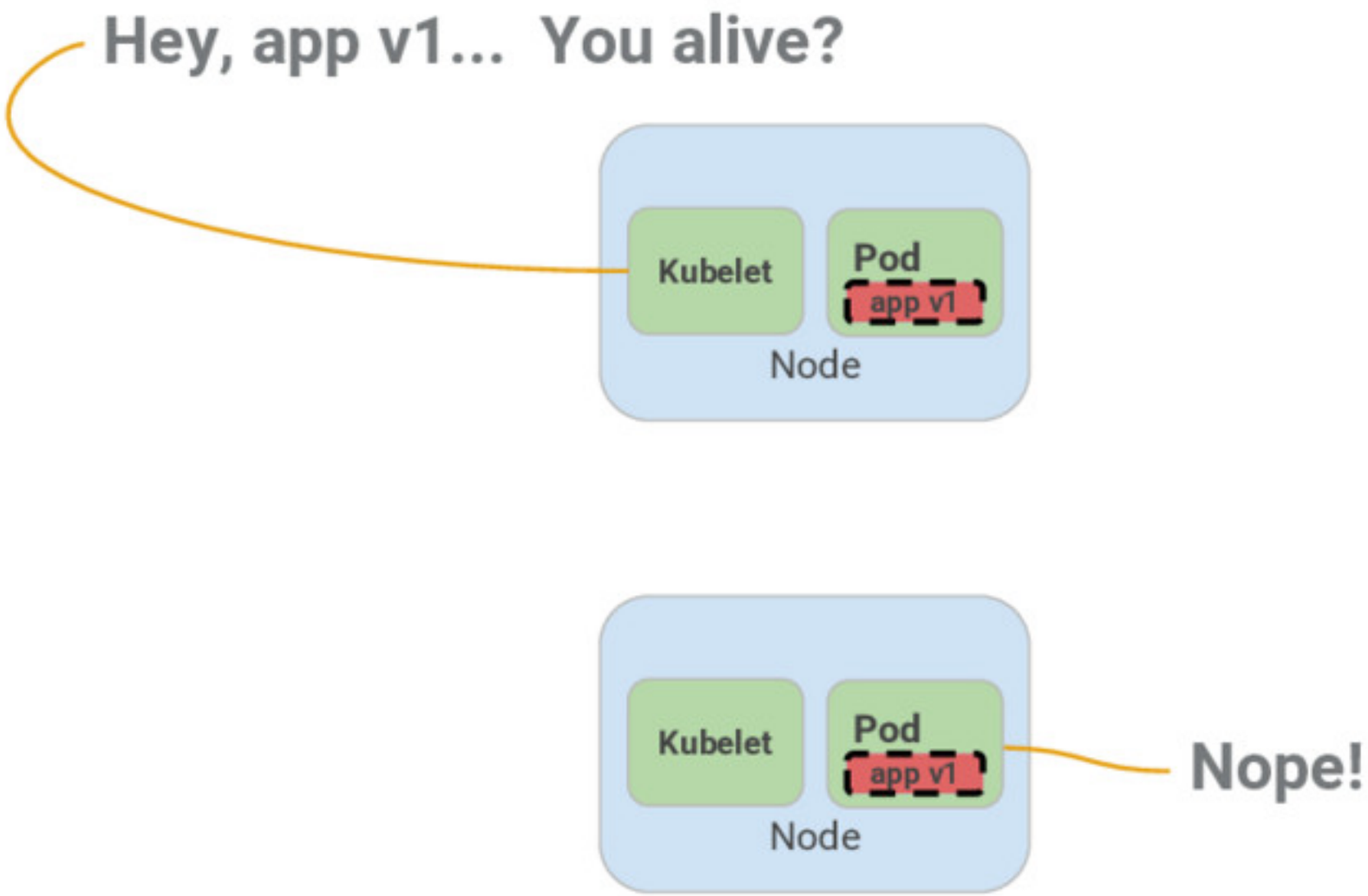
ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Monitoring and Health Checks

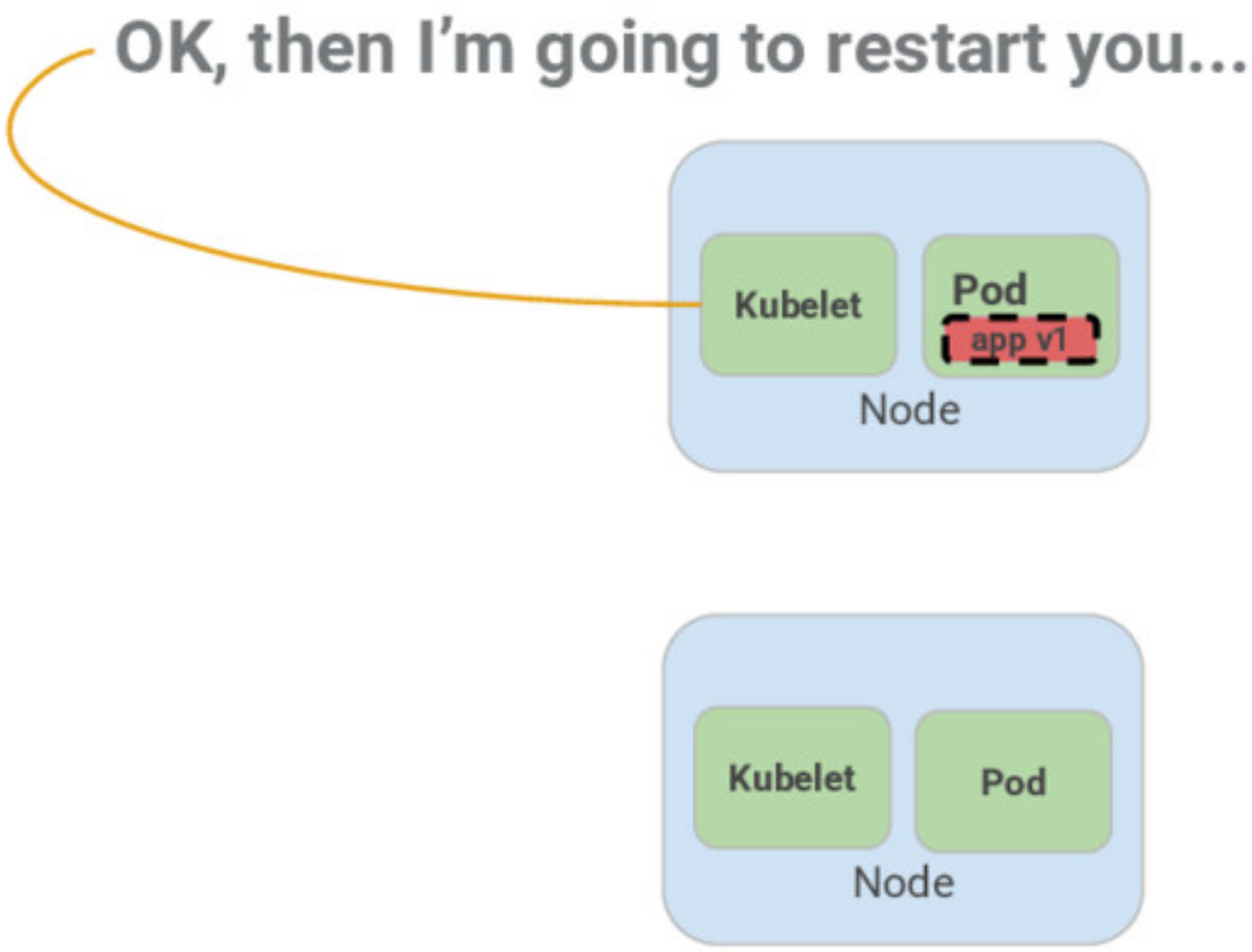


- On every node is a daemon called a Kubelet
- One of the Kubelete's jobs is to ensure that pods are healthy

Monitoring and Health Checks



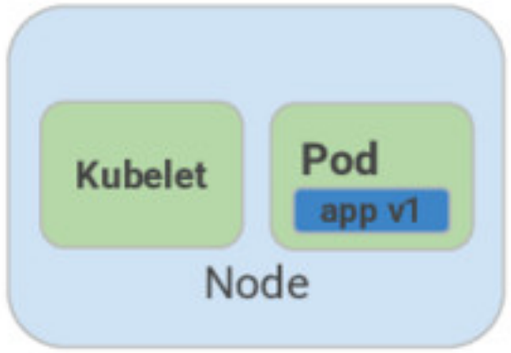
Monitoring and Health Checks



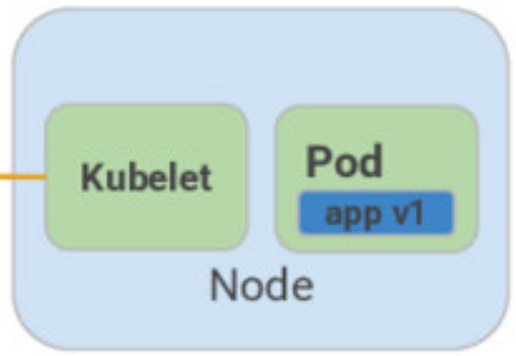
P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

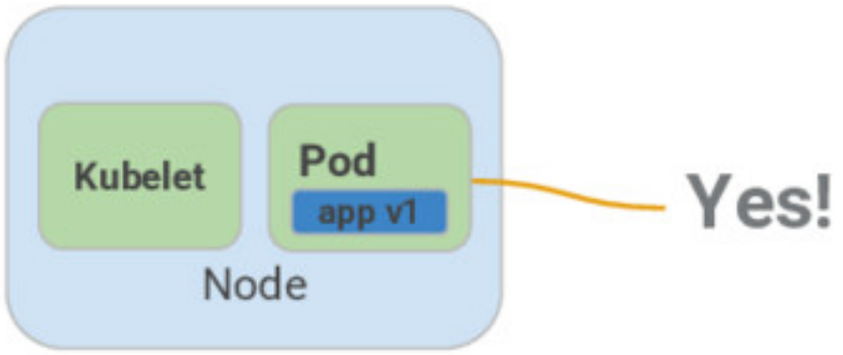
Monitoring and Health Checks



Hey, app v1... You alive?



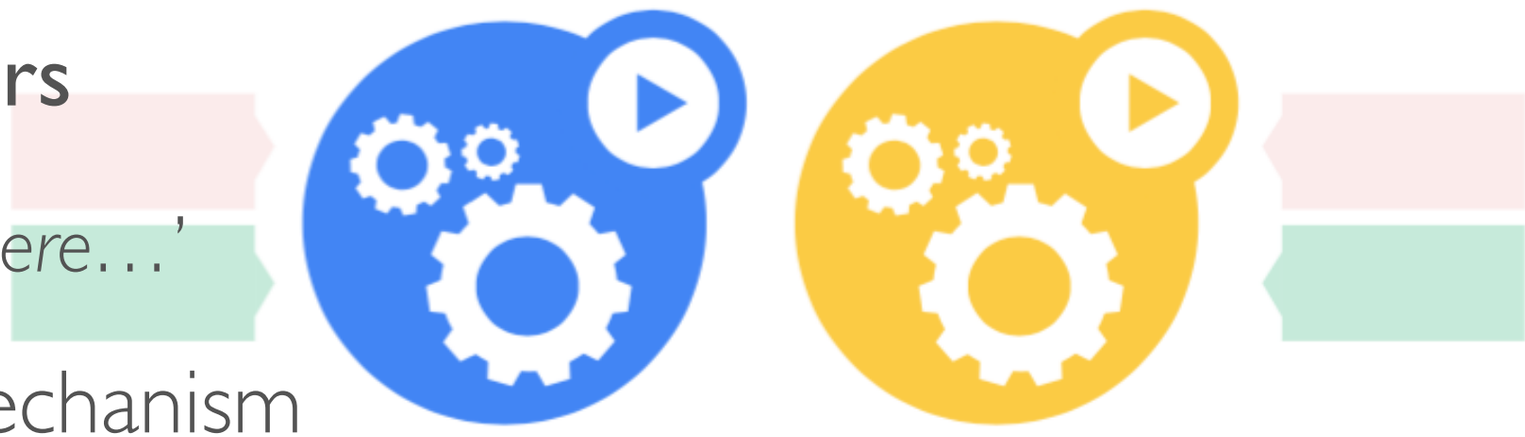
Monitoring and Health Checks



Labels

- Arbitrary metadata
- Attached to any API object

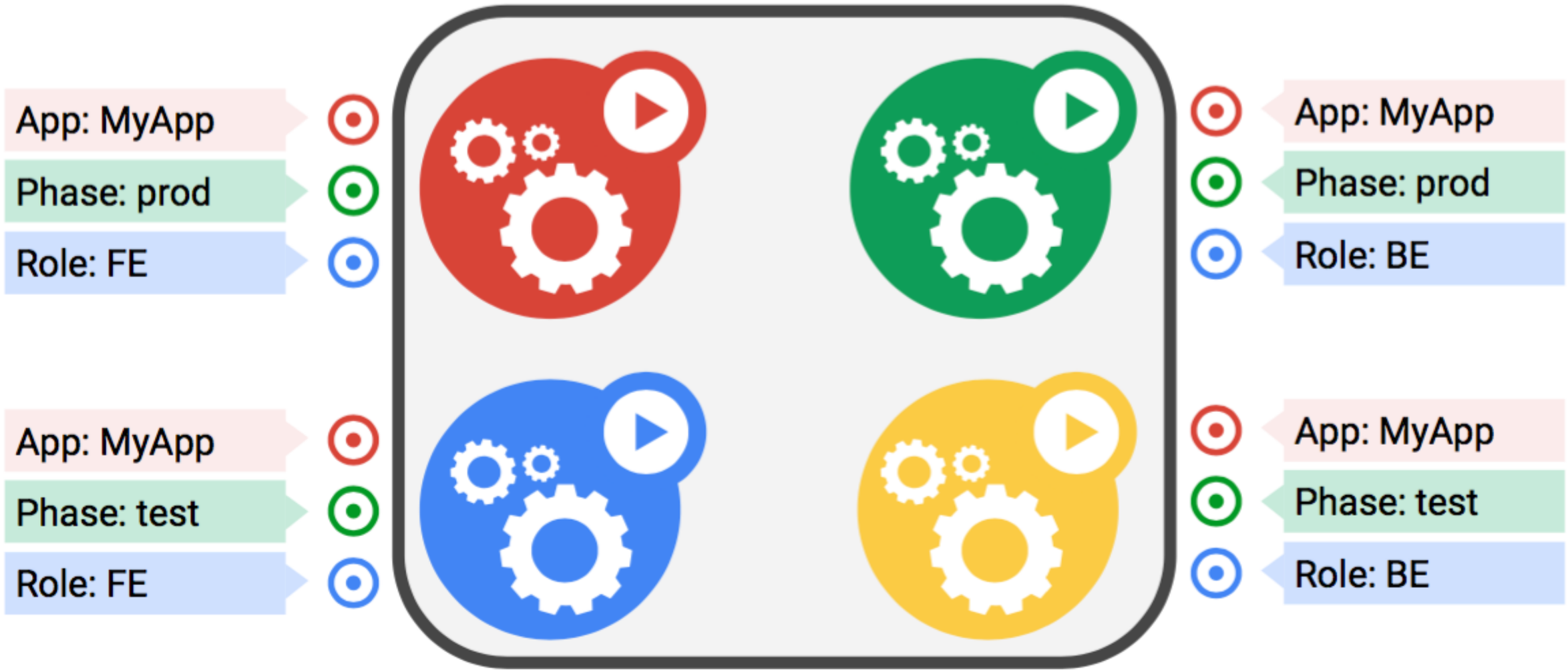
- Generally represent **identity**
- Queryable by **selectors**
  - think SQL '*select*… *where*…'
- The **only** grouping mechanism
  - pods under a ReplicaSet
  - pods in a Service
  - capabilities of a node (constraints)



Selectors

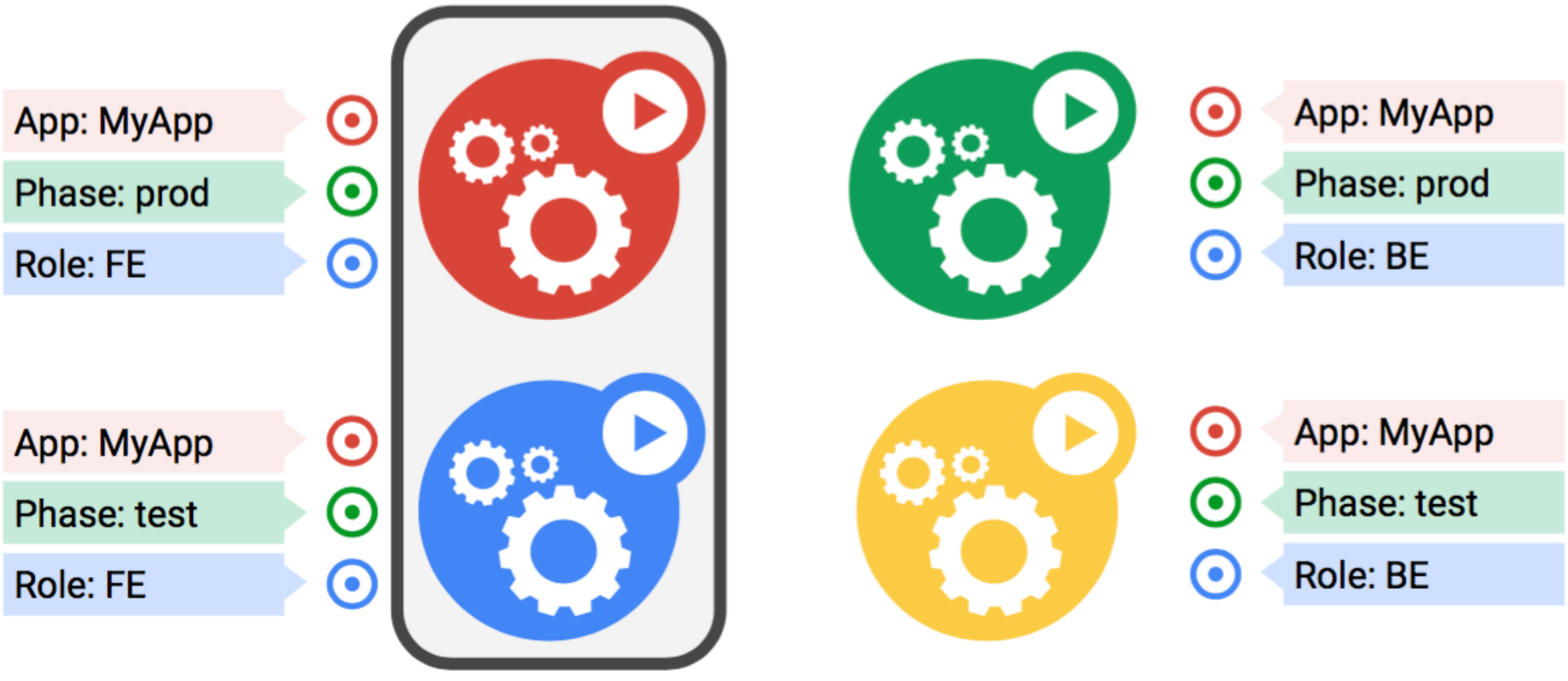

App: MyApp
Phase: prod
Role: FE
App: MyApp
Phase: prod
Role: BE
App: MyApp
Phase: test
Role: FE
App: MyApp
Phase: test
Role: BE

Selectors



App = MyApp

Selectors

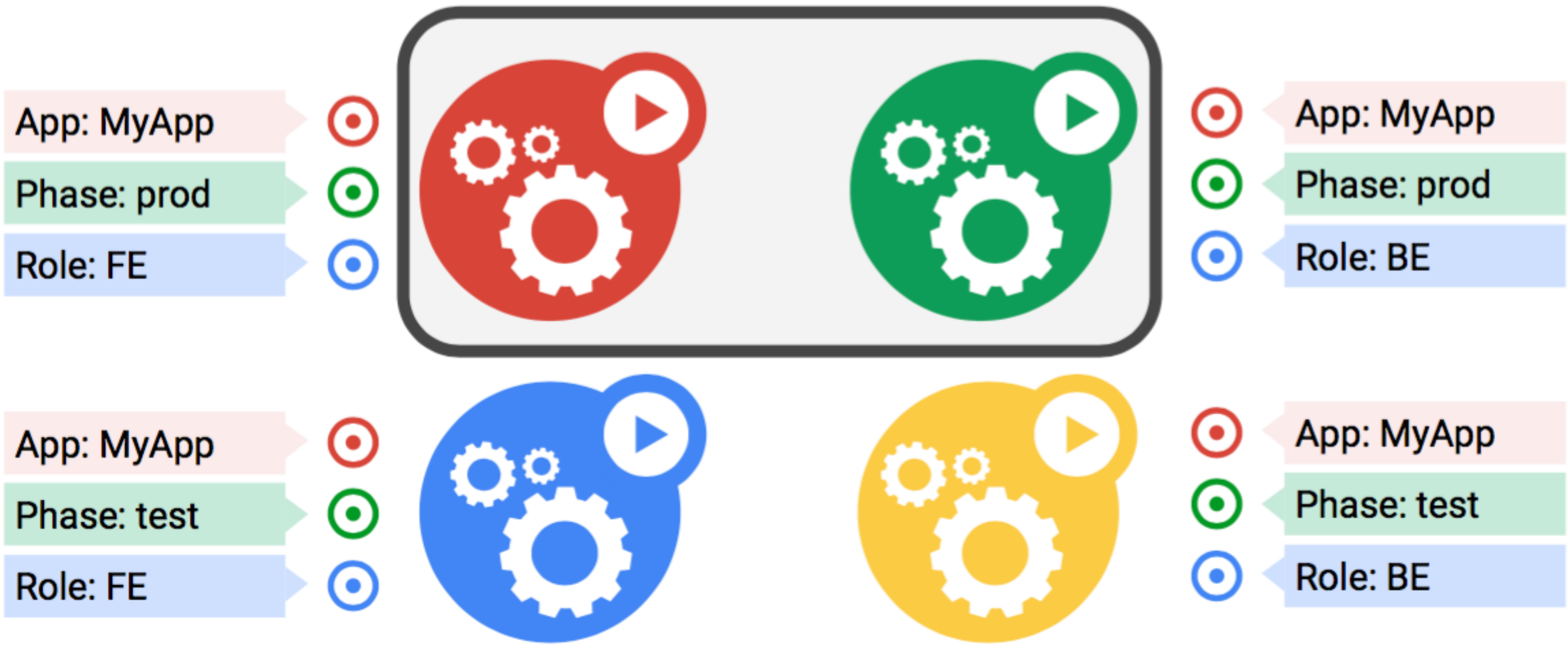


App = MyApp, Role = FE

Selectors

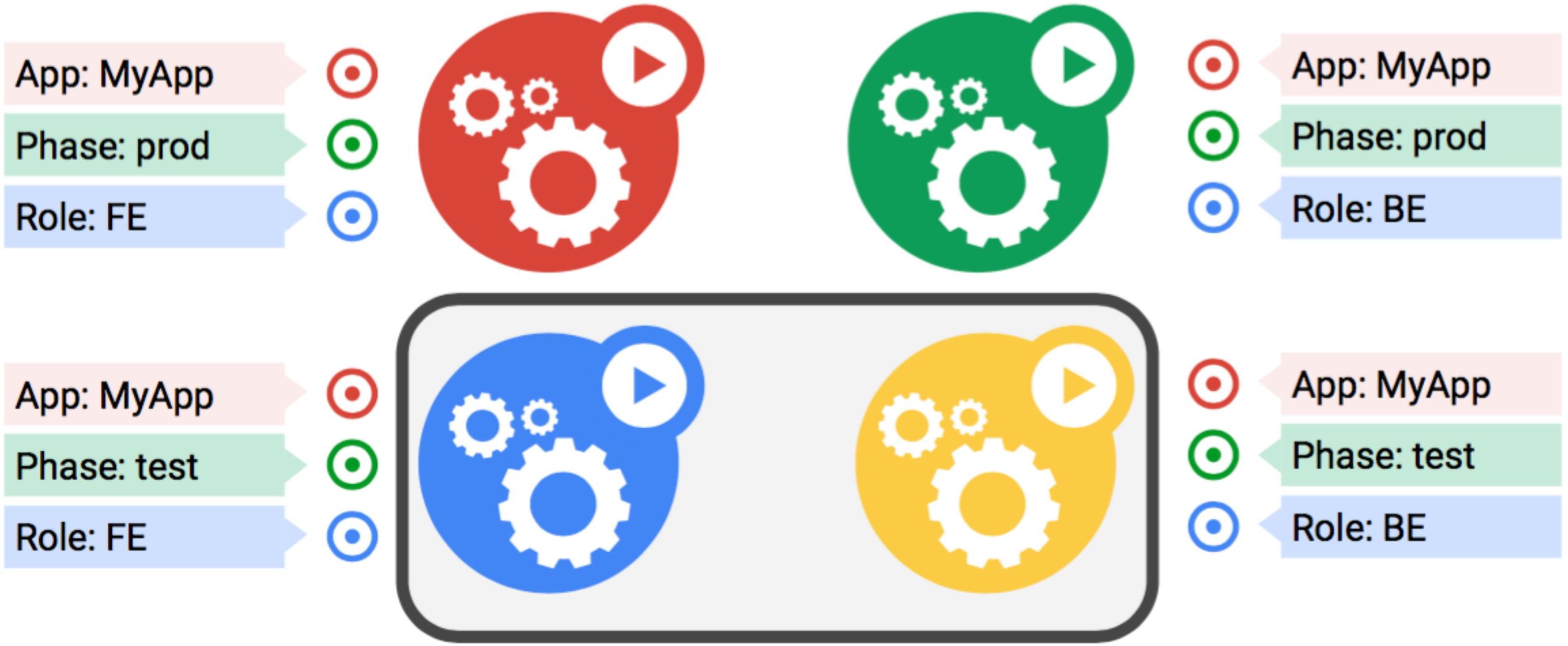
App = MyApp, Role = BE

Selectors



App = MyApp, Phase = prod

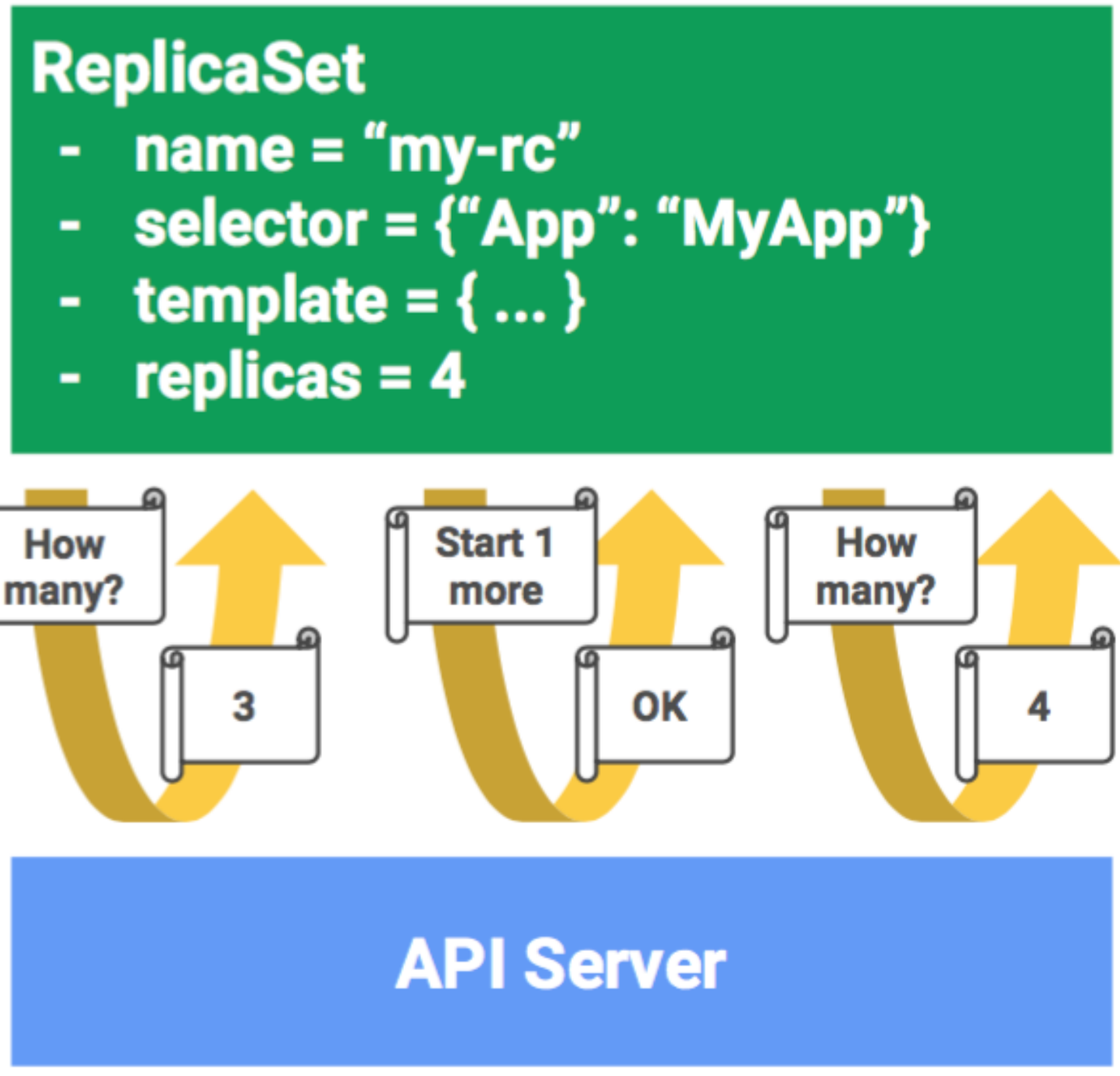
Selectors



App = MyApp, Phase = test

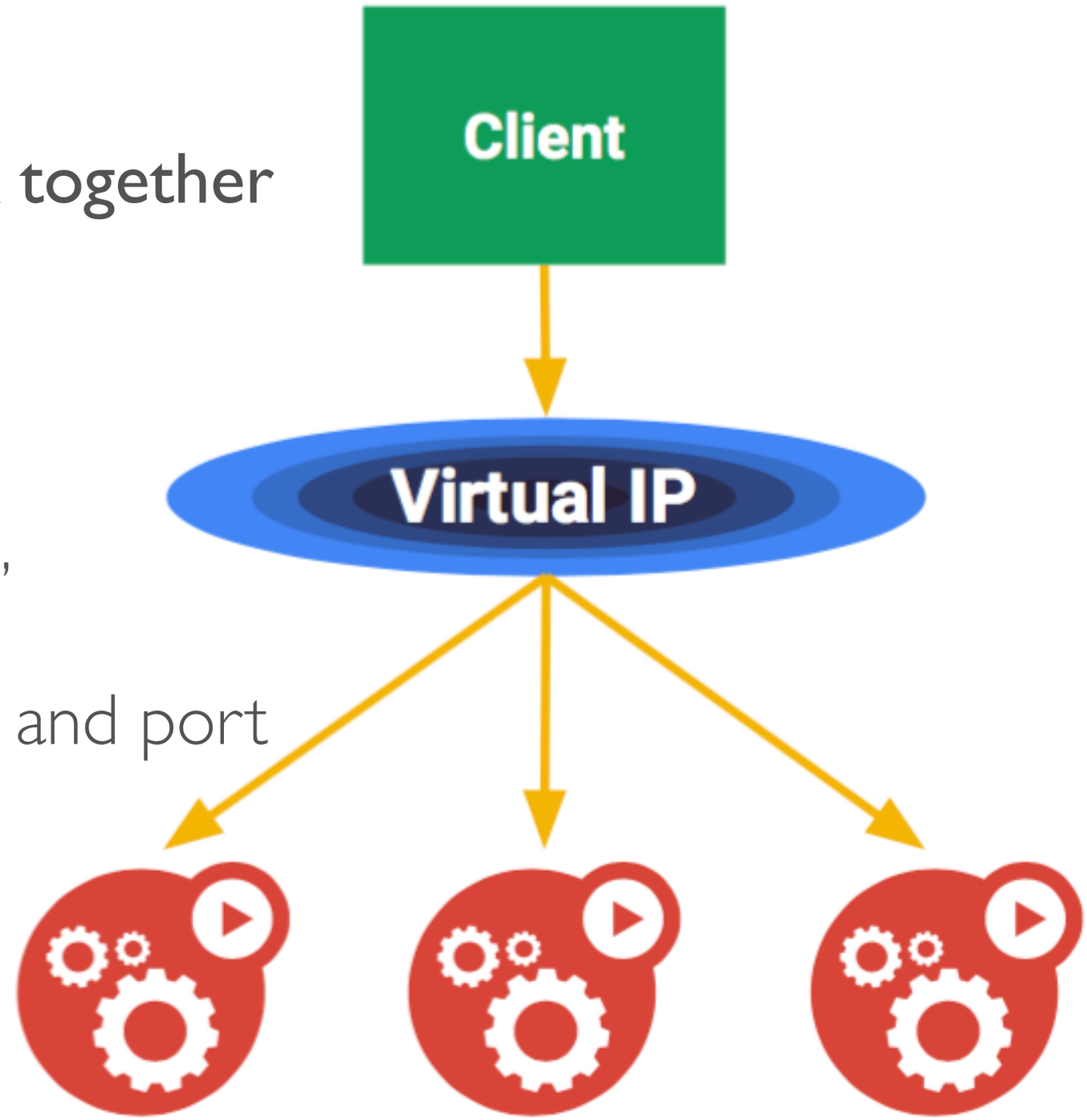
ReplicaSets

- A simple control loop
- **One job**: ensure N copies of a pod
  - grouped by a selector
  - too few? start some
  - too many? kill some



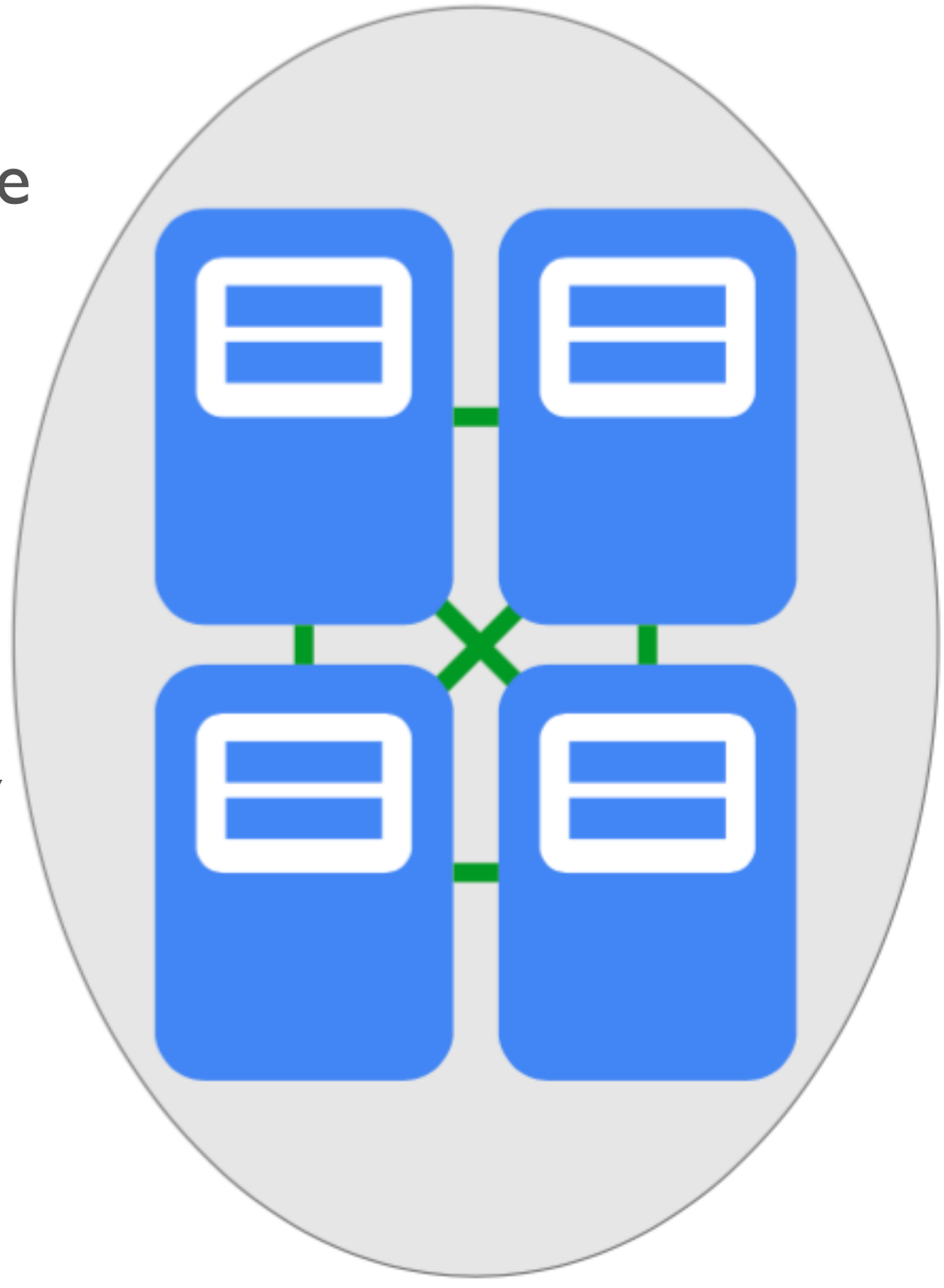
Services

- A group of **pods that work together**
  - grouped by a selector
- Defines access policy
  - "load balanced" or "headless"
- Can have a stable **virtual IP** and port
  - also a DNS name
- **Hides complexity**



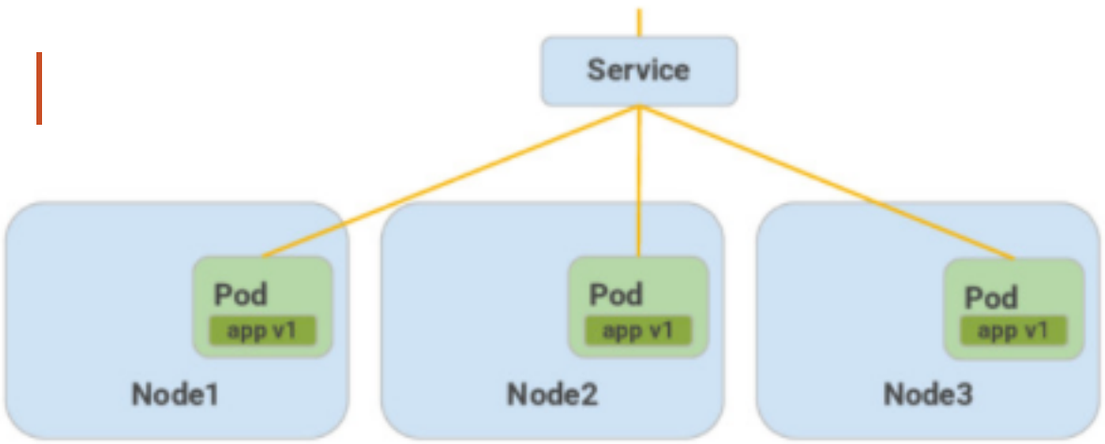
External Services

- Services VIPs are only available **inside** the cluster
- Need to receive traffic from "the outside world"
- Service "type"
  - NodePort: expose on a port on every node
  - LoadBalancer: provision a cloud load-balancer

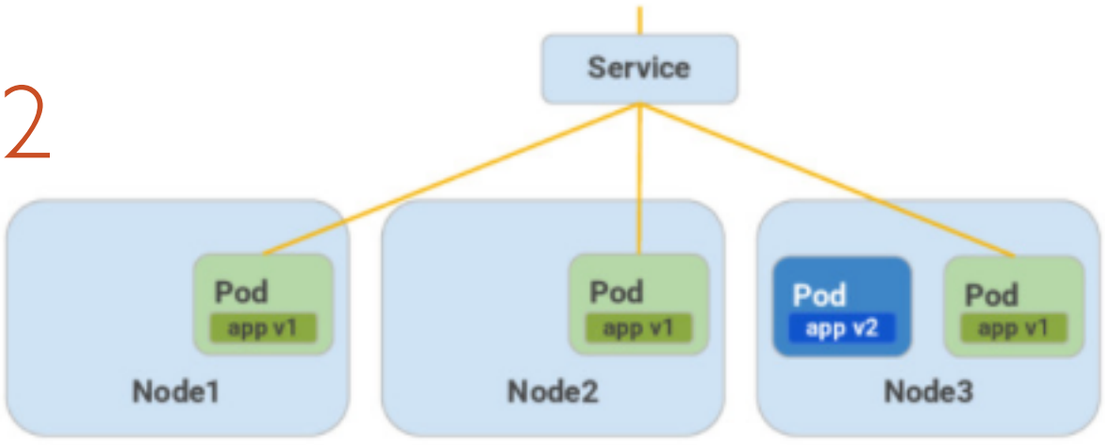


Rolling Update

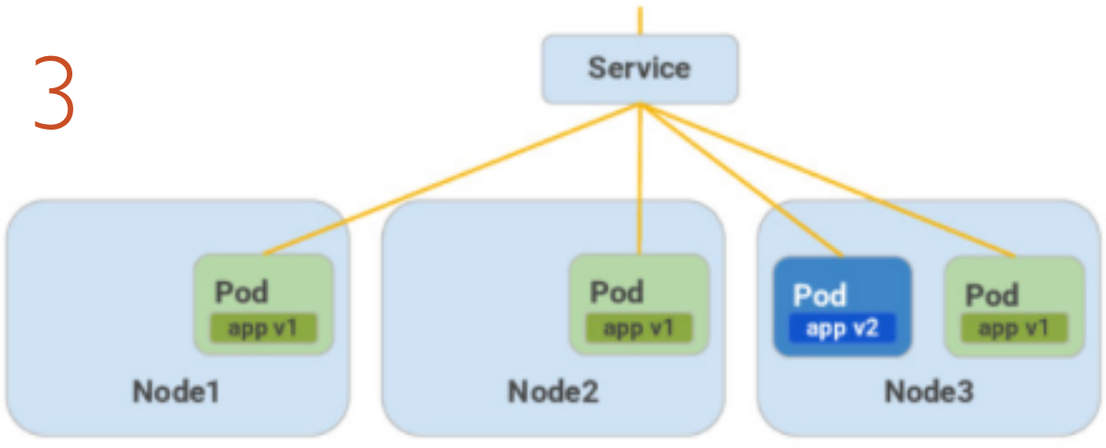
1



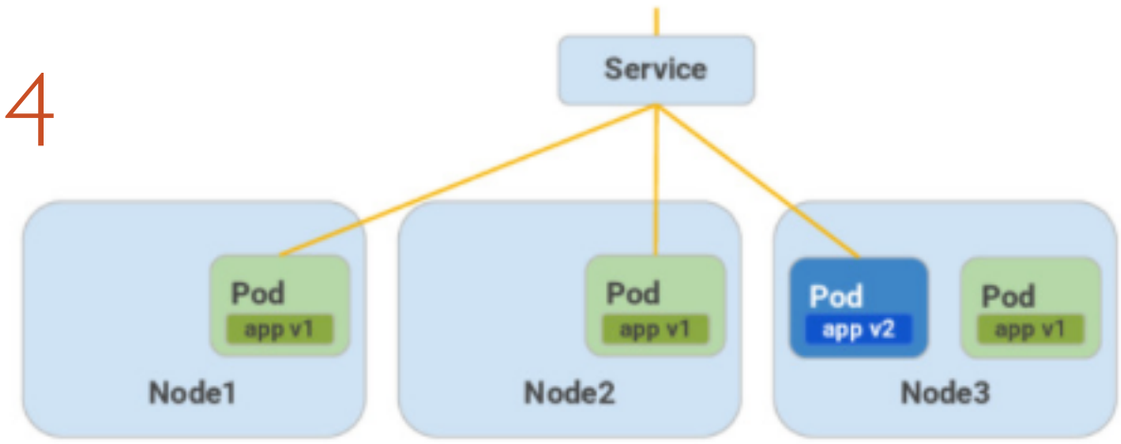
2



3

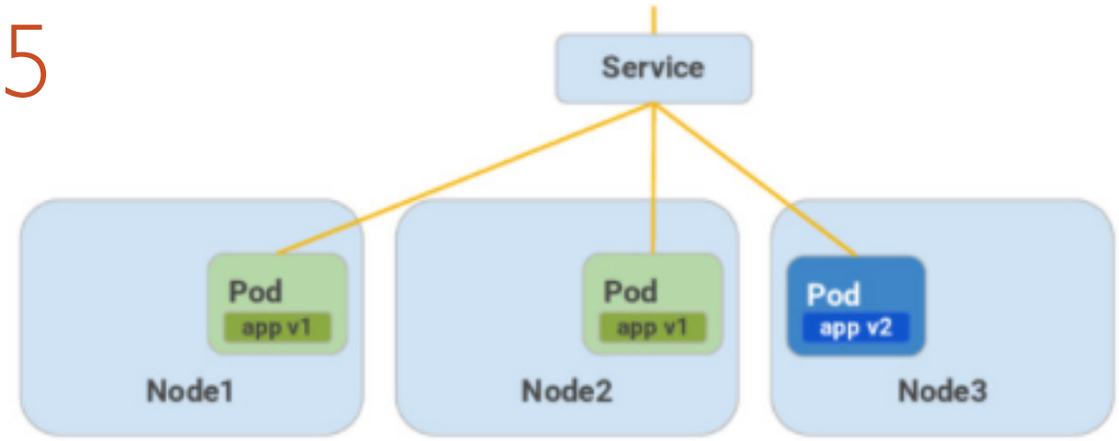


4

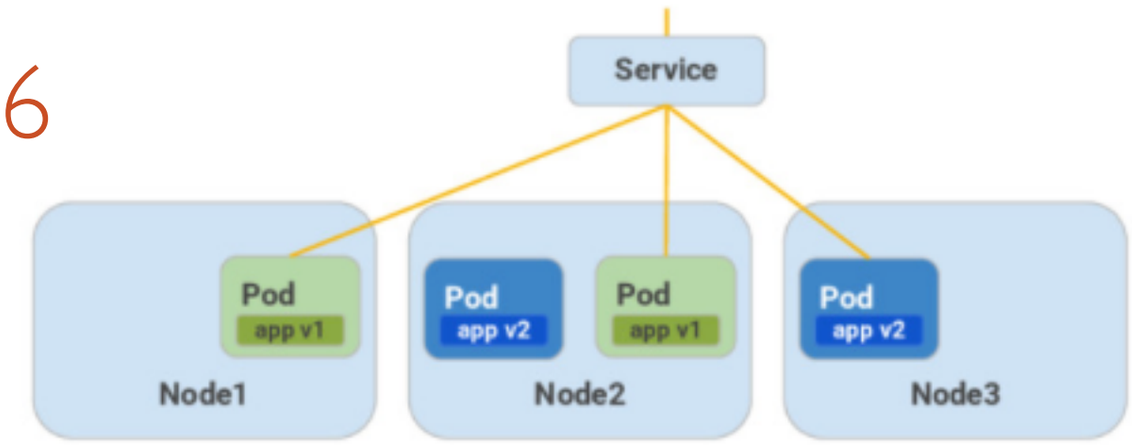


Rolling Update

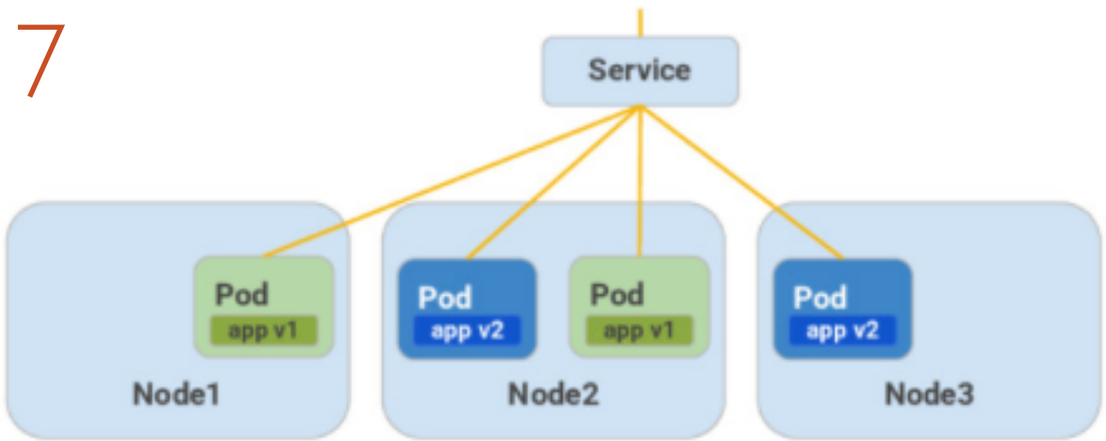
5



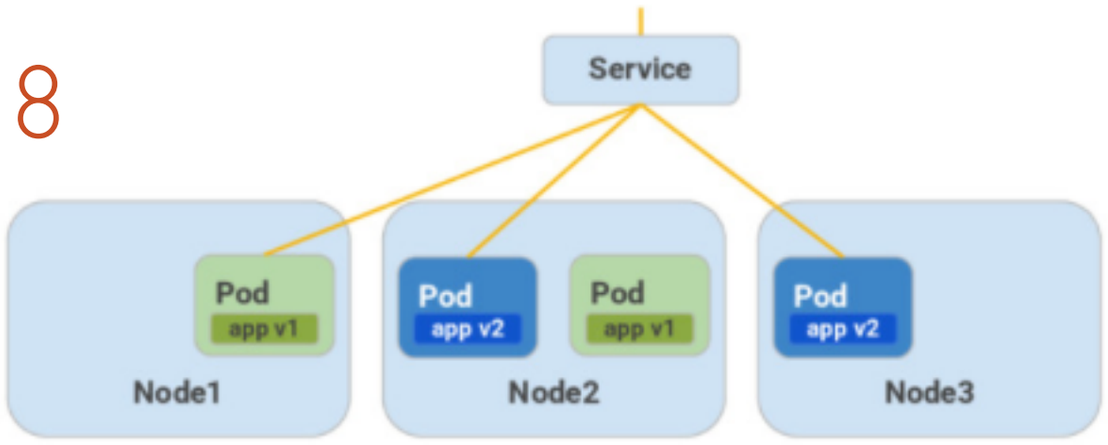
6



7

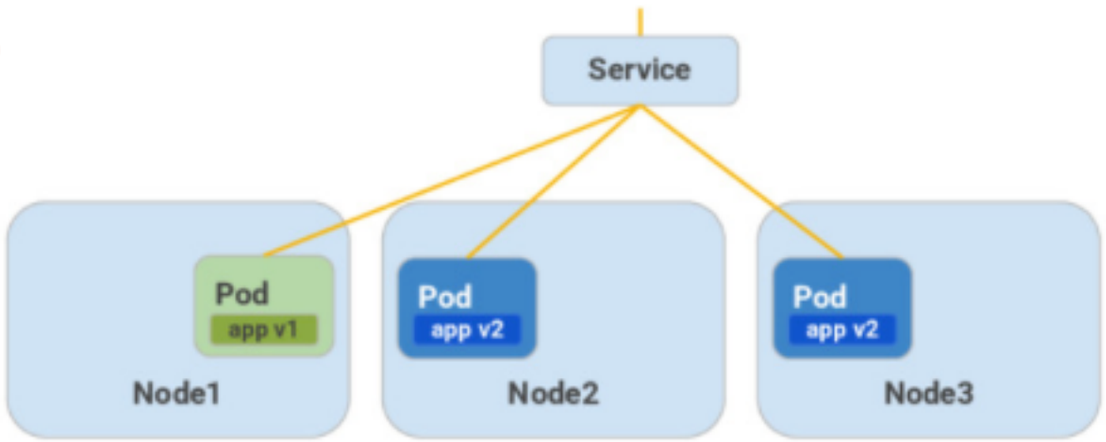


8

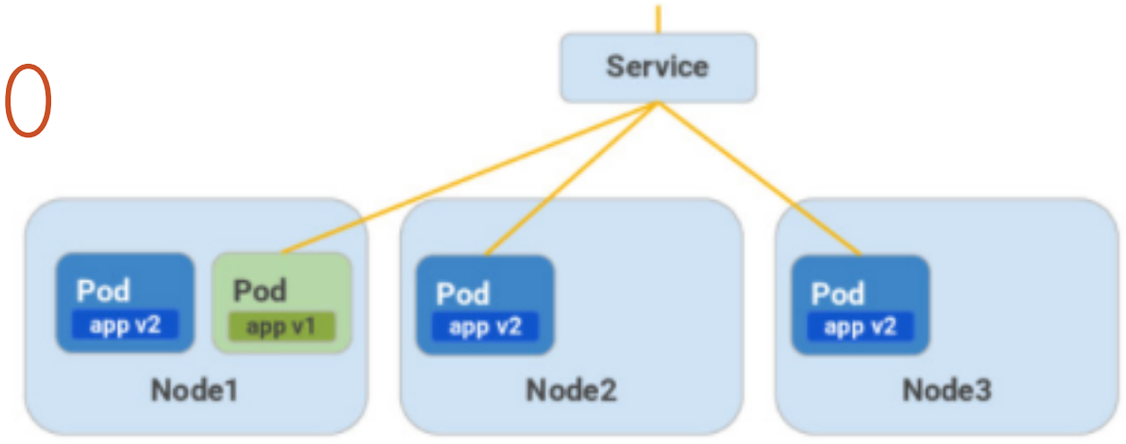


Rolling Update

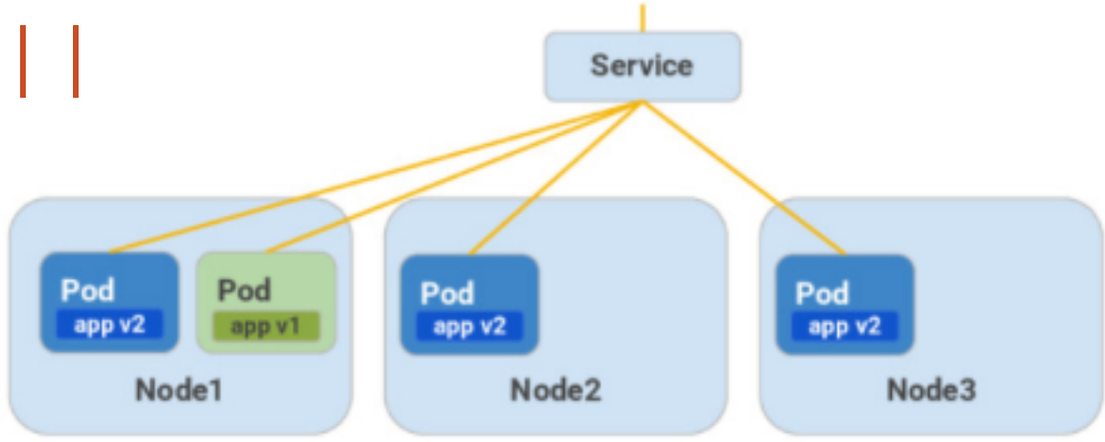
9



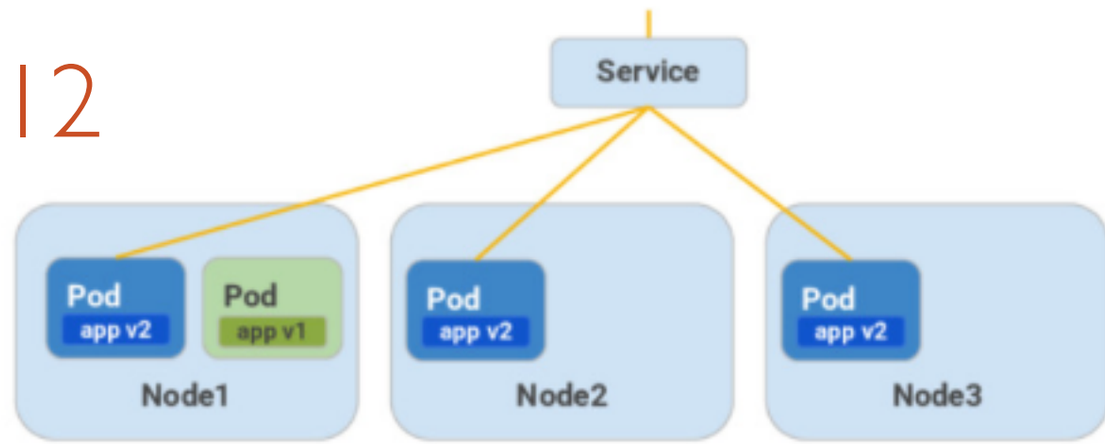
10



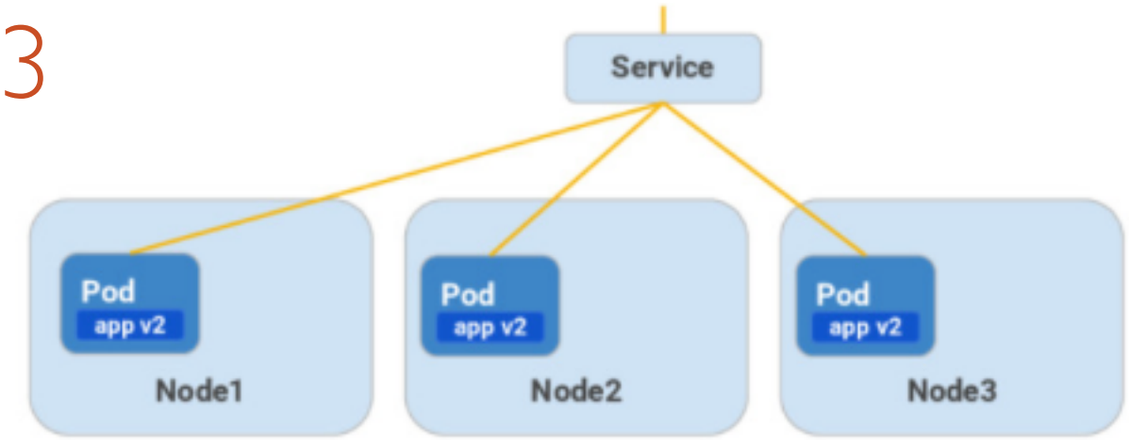
11



12



Rolling Update



kubernetes

Hands on lab

Welcome to Kubernetes

- Minikube - Intended for local kubernetes development
- Docker

https://kubernetes.io/docs/tasks/tools/install-minikube/

1. Install **kubectl** binary with **curl** (Linux and MacOS)
2. On Windows download and add the folder to the path
3. Try the the tool on the terminal (on windows, install a different command line) - **`kubectl`**

```
~                                                                    ✔ 41  10:48:10
$ kubectl
kubectl controls the Kubernetes cluster manager.

 Find more information at:
https://kubernetes.io/docs/reference/kubectl/overview/

Basic Commands (Beginner):
  create         Create a resource from a file or from stdin.
  expose         Take a replication controller, service, deployment or pod and
expose it as a new Kubernetes Service
  run            Run a particular image on the cluster
  set            Set specific features on objects

Basic Commands (Intermediate):
  explain        Documentation of resources
  get            Display one or many resources
  edit           Edit a resource on the server
  delete         Delete resources by filenames, stdin, resources and names, or
by resources and label selector
```

1. Install **minikube** binary with **`curl`** (Linux and MacOS)
2. On Windows download and add the folder where you downloaded **kubectl**
3. Try the the tool on the terminal (on windows, install a different command line) - **`minikube`**

```
~                                                                    ✔ 44  10:50:39
$ minikube
Minikube is a CLI tool that provisions and manages single-node Kubernetes clusters optimized for
development workflows.

Basic Commands:
  start          Starts a local kubernetes cluster
  status         Gets the status of a local kubernetes cluster
  stop           Stops a running local kubernetes cluster
  delete         Deletes a local kubernetes cluster
  dashboard      Access the kubernetes dashboard running within the minikube cluster

Images Commands:
  docker-env     Sets up docker env variables; similar to '$(docker-machine env)'
  cache          Add or delete an image from the local cache.
```

Try minikube

1. On linux and MacOS start **minikube - `minikube start`**

```
~                                                                    ✔ 44 10:50:43
$ minikube start
😄  minikube v1.4.0 on Darwin 10.14.6
💡  Tip: Use 'minikube start -p <name>' to create a new cluster, or 'minikube delete' to delete this o
ne.
🔄  Starting existing virtualbox VM for "minikube" ...
⌛  Waiting for the host to be provisioned ...
🐳  Preparing Kubernetes v1.16.0 on Docker 18.09.9 ...
🔄  Relaunching Kubernetes using kubeadm ...
⌛  Waiting for: apiserver proxy etcd scheduler controller dns
🏄  Done! kubectl is now configured to use "minikube"
```

2. On Windows (with **HyperV**)

```
~                                                                    ✔ 47 11:27:25
$ minikube start --vm-driver="hyperv" --hyperv-virtual-switch="switch"
```

kubernetes

3. Kubectl version

```
~                                                                    ✔ 47 11:27:25
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"16", GitVersion:"v1.16.0", GitCommit:"2bd9643cee5b3b3a5
ecbd3af49d09018f0773c77", GitTreeState:"clean", BuildDate:"2019-09-18T14:36:53Z", GoVersion:"go1.12.9"
, Compiler:"gc", Platform:"darwin/amd64"}
Server Version: version.Info{Major:"1", Minor:"16", GitVersion:"v1.16.0", GitCommit:"2bd9643cee5b3b3a5
ecbd3af49d09018f0773c77", GitTreeState:"clean", BuildDate:"2019-09-18T14:27:17Z", GoVersion:"go1.12.9"
, Compiler:"gc", Platform:"linux/amd64"}
```

There should be a Client and Server version

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

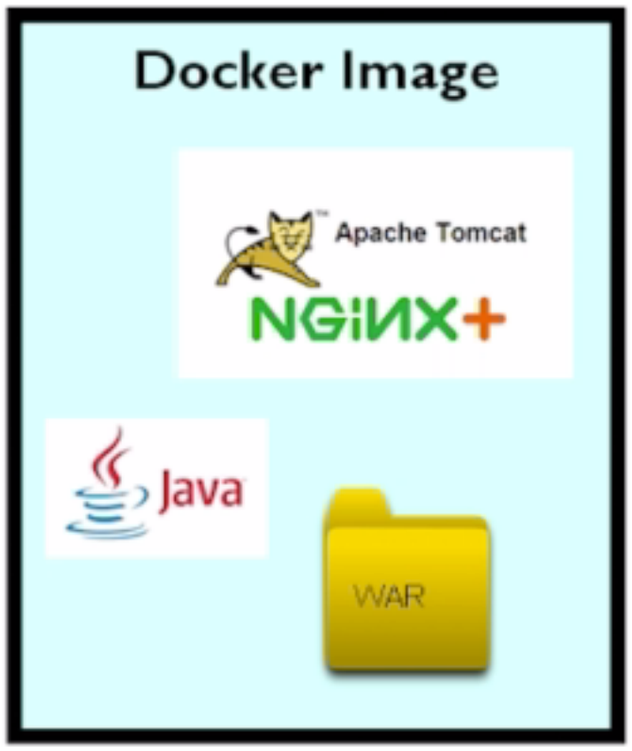
Docker: Important things...

- Most important concept in docker

containers vs images

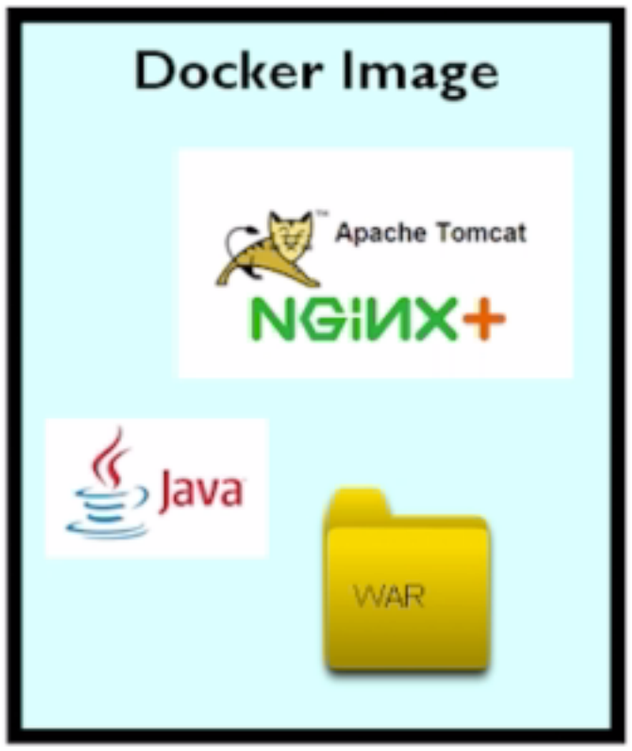
- An image e a definition of a container
- binary file with all the software a things like ambiente variables, general definitions, etc

Let's imagine i'm a developer given the task to do a website



In the end I would deliver the war file to someone responsible to deploy it

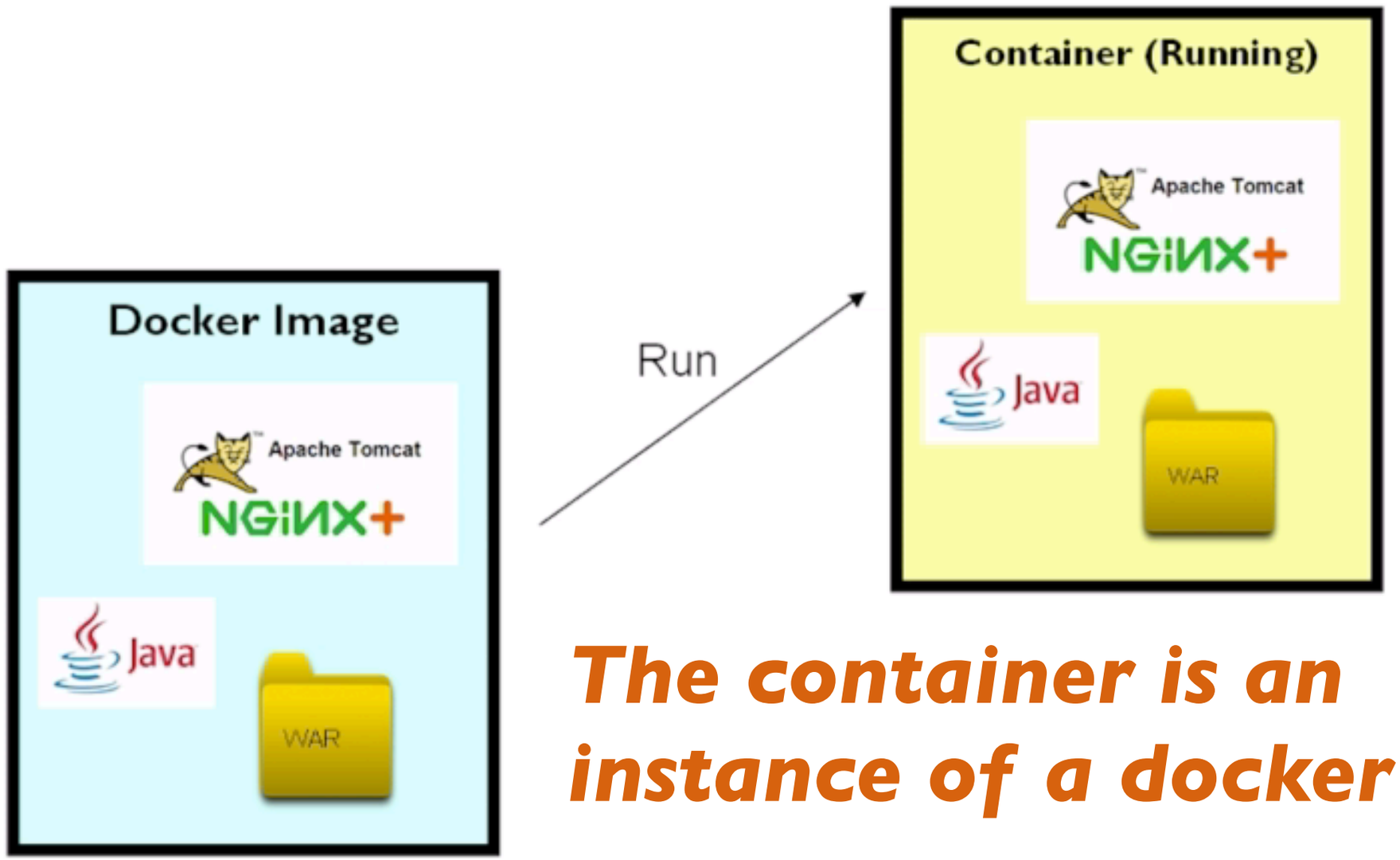
will it work?



- I would expect them to configure **java**, **tomcat** and **nginx**

With an image, I as a developer can package everything that is needed every single dependency required to make that application run

- The idea then is to pass the image to the deployer - then he has to run that image
- The running image is the **container**

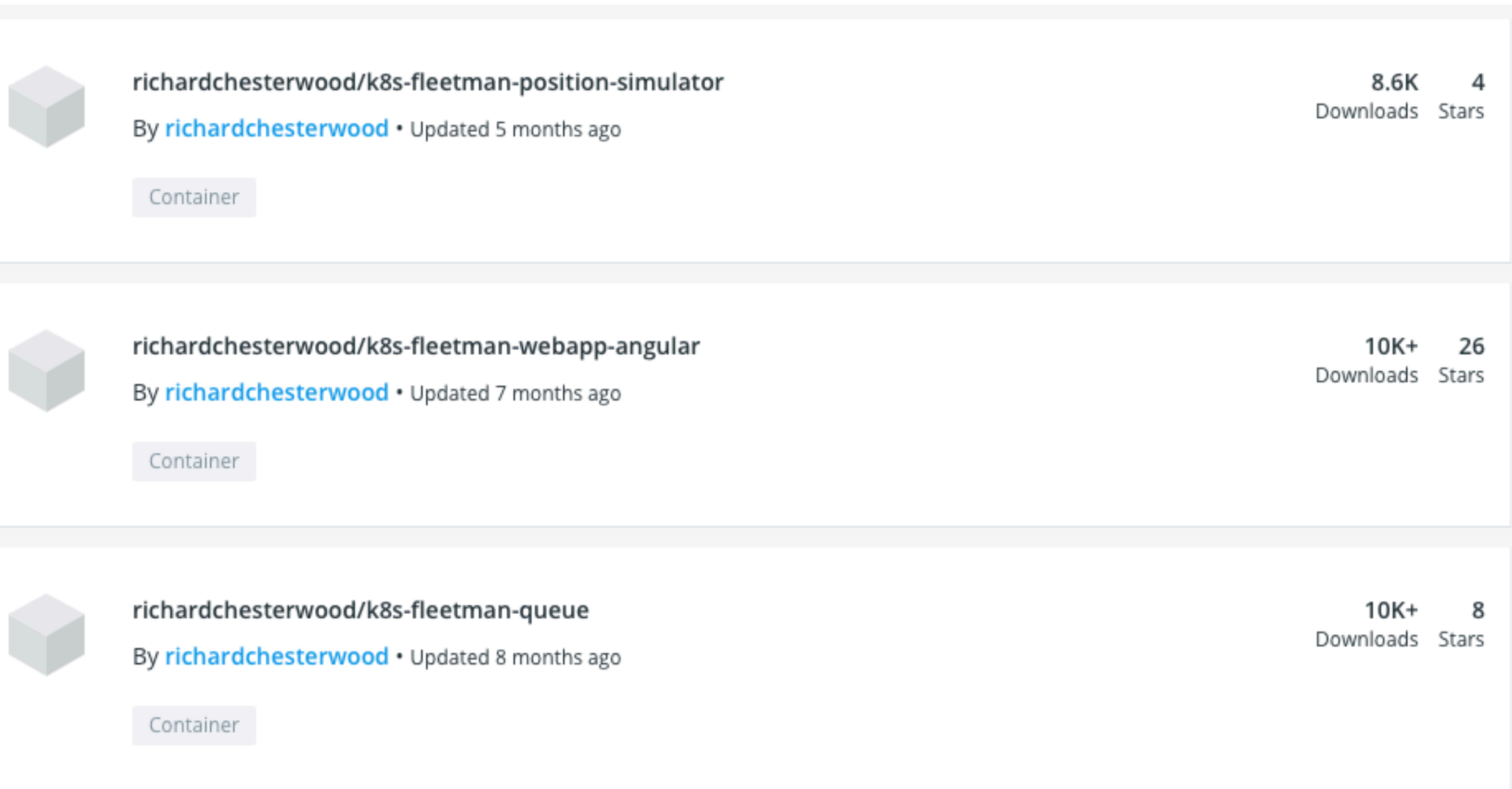


The container is an instance of a docker image

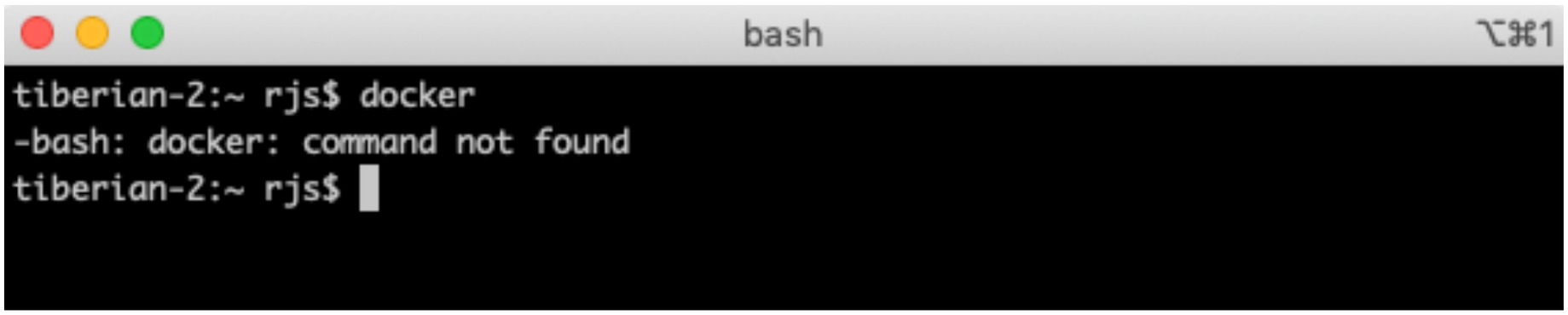
- We can publish our images on the docker ecosystem for other developers

`hub.docker.com`

- We will use for the next lessons the following repository



- You will need docker installed



- If you are in a Mac - get **Docker for Mac**
- Otherwise, get **Docker Toolbox**

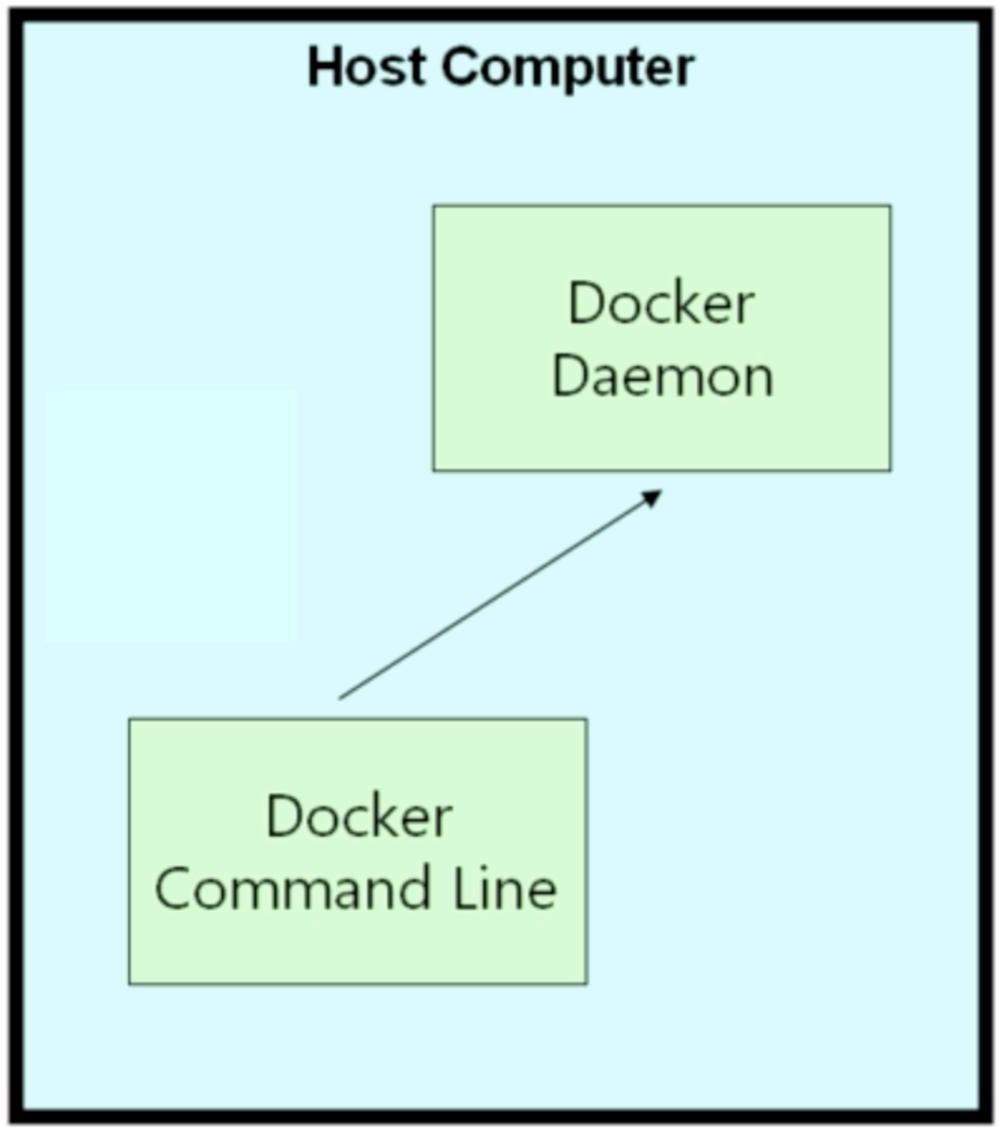
- If you are in a Mac - get Docker for Mac
- Otherwise, get Docker Toolbox

```
~
$ docker                                                    ✔ 49 11:37:20

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default "/Users/rjs/.docker")
  -c, --context string     Name of the context to use to connect to the daemon (overrides
                           DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
```

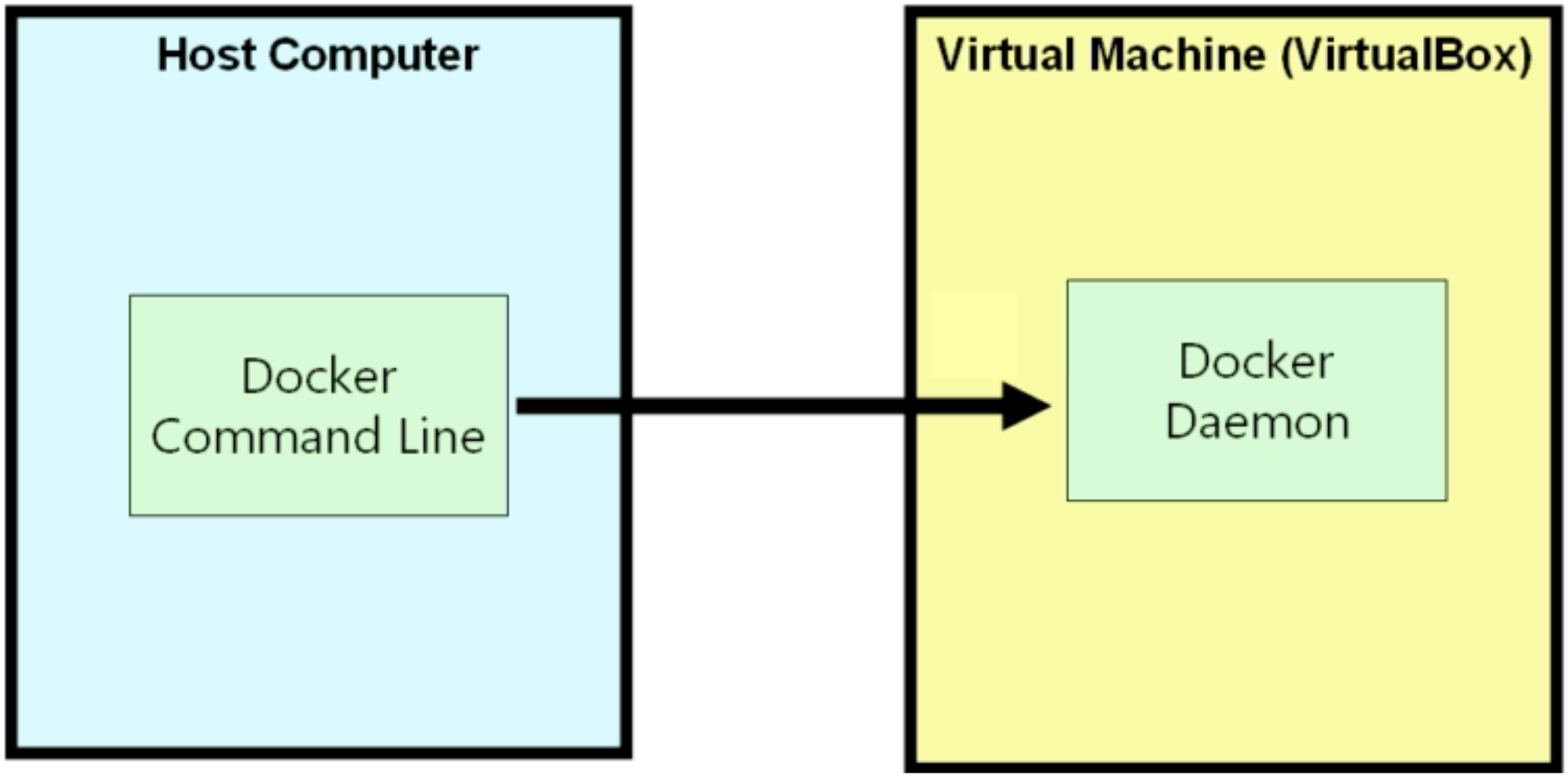
- It took a lot of time right?

```
~                                                                    ✔ 51 11:39:36
$ docker image ls
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

~                                                                    1 ↵ 52 11:39:39
$
```

Why?

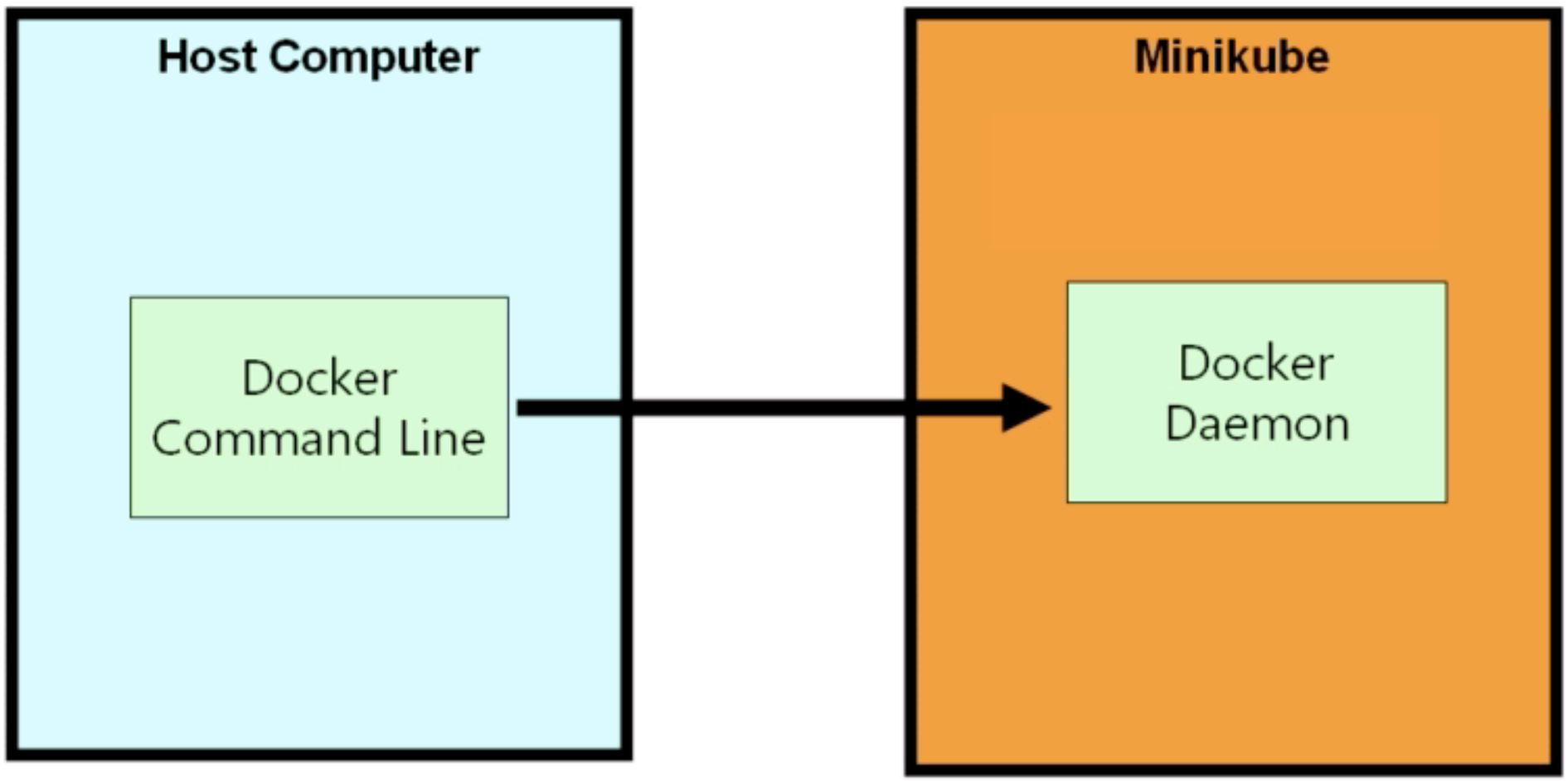


- Docker daemon need to run on linux - because it uses various features of the linux operation system to support containers



- Docker tries to look for a virtual machine - looks for a docker daemon on the VM

- You have to start that VM
- Because we intend to work on Kubernetes, and we installed Minikube (cut down version of kubernetes)
- Inside Minikube we have an installation of docker - docker daemon


Host Computer
Docker
Command Line
Minikube
Docker
Daemon

- Let's make use of that docker installation - `minikube start`

```
~                                                                    ✔ 44 10:50:43
$ minikube start
😄  minikube v1.4.0 on Darwin 10.14.6
💡  Tip: Use 'minikube start -p <name>' to create a new cluster, or 'minikube delete' to delete this o
ne.
🔄  Starting existing virtualbox VM for "minikube" ...
⌛  Waiting for the host to be provisioned ...
🐳  Preparing Kubernetes v1.16.0 on Docker 18.09.9 ...
🔄  Relaunching Kubernetes using kubeadm ...
⌛  Waiting for: apiserver proxy etcd scheduler controller dns
🏄  Done! kubectl is now configured to use "minikube"
```

- `docker image ls`

```
~                                                                    ✔ 51 11:39:36
$ docker image ls
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

~                                                                 1 ↵ 52 11:39:39
$
```

- We need to configure docker to tell it that we are talking to the Minikube environment rather that the tradition docker running on its virtual machine - **`minikube docker-env`**

```
~
$ minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/Users/rjs/.minikube/certs"
# Run this command to configure your shell:
# eval $(minikube docker-env)
```

✔ 53 12:41:19

Environment variable needed to setup in order to use the docker inside Minikube

```
~
$ eval $(minikube docker-env)
```

✔ 55 12:44:14

```
~                                                          ✔ 59 15:08:09
$ echo $DOCKER_HOST
tcp://192.168.99.100:2376
```

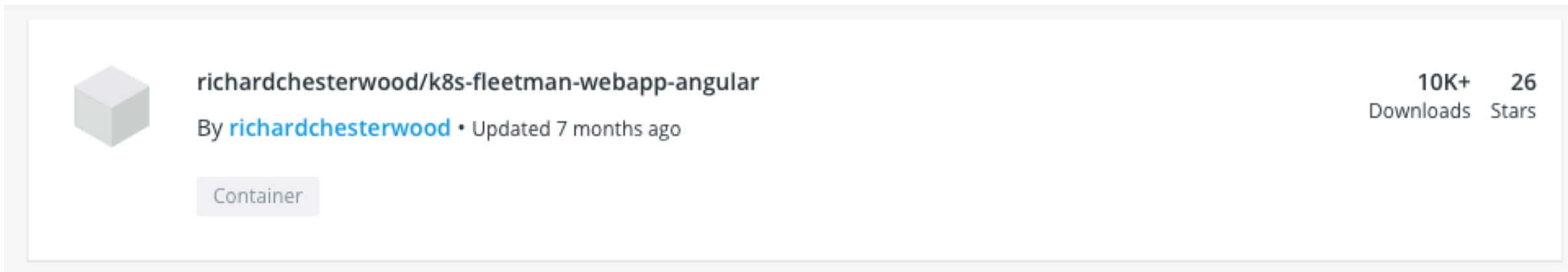
```
~                                                          ✔ 53 12:41:19
$ minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/Users/rjs/.minikube/certs"
# Run this command to configure your shell:
# eval $(minikube docker-env)
```

```
~                                                                    ✔ 62 15:10:26
$ docker image ls
REPOSITORY                                TAG                 IMAGE ID            CREATED
       SIZE
k8s.gcr.io/kube-apiserver                 v1.16.0             b305571ca60a        4 weeks ago
       217MB
k8s.gcr.io/kube-controller-manager        v1.16.0             06a629a7e51c        4 weeks ago
       163MB
k8s.gcr.io/kube-proxy                     v1.16.0             c21b0c7400f9        4 weeks ago
       86.1MB
k8s.gcr.io/kube-scheduler                 v1.16.0             301ddc62b80b        4 weeks ago
       87.3MB
```

If you don't use linux this configuration is much better in term of performance

- This ways you don't have two VMs running (docker toolbox + minikube)

- Go to **`hub.docker.com`** and find the following image



- How can we deploy this?
- First we need to see the releases

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

release0 9.76 MB

Last updated **a year ago** by richardchesterwood

DIGEST	ARCHITECTURE	OS	SIZE
9b98fec20772	amd64	linux	9.76 MB

release0-5 9.76 MB

Last updated **a year ago** by richardchesterwood

DIGEST	ARCHITECTURE	OS	SIZE
636f57ab20c3	amd64	linux	9.76 MB

release2 25.07 MB

Last updated **a year ago** by richardchesterwood

DIGEST	ARCHITECTURE	OS	SIZE
ed7d720878ac	amd64	linux	25.07 MB

release1 25.07 MB

Last updated **a year ago** by richardchesterwood

DIGEST	ARCHITECTURE	OS	SIZE
46bef951fc3c	amd64	linux	25.07 MB

- First download the image for our computer (optional step)

```
~
$ docker image pull richardchesterwood/k8s-fleetman-webapp-angular:release0-5
                                                                    ✔ 73 17:09:06
```

```
~
$ docker image pull richardchesterwood/k8s-fleetman-webapp-angular:release0-5
release0-5: Pulling from richardchesterwood/k8s-fleetman-webapp-angular
Digest: sha256:636f57ab20c367bcab78d1994d4990b72bc8f69686fc716bc06386c2894fb8dd
Status: Image is up to date for richardchesterwood/k8s-fleetman-webapp-angular:release0-5
docker.io/richardchesterwood/k8s-fleetman-webapp-angular:release0-5
                                                                    ✔ 73 17:09:06
```

- You should see it in your images list - `docker image ls`

Let's run it!

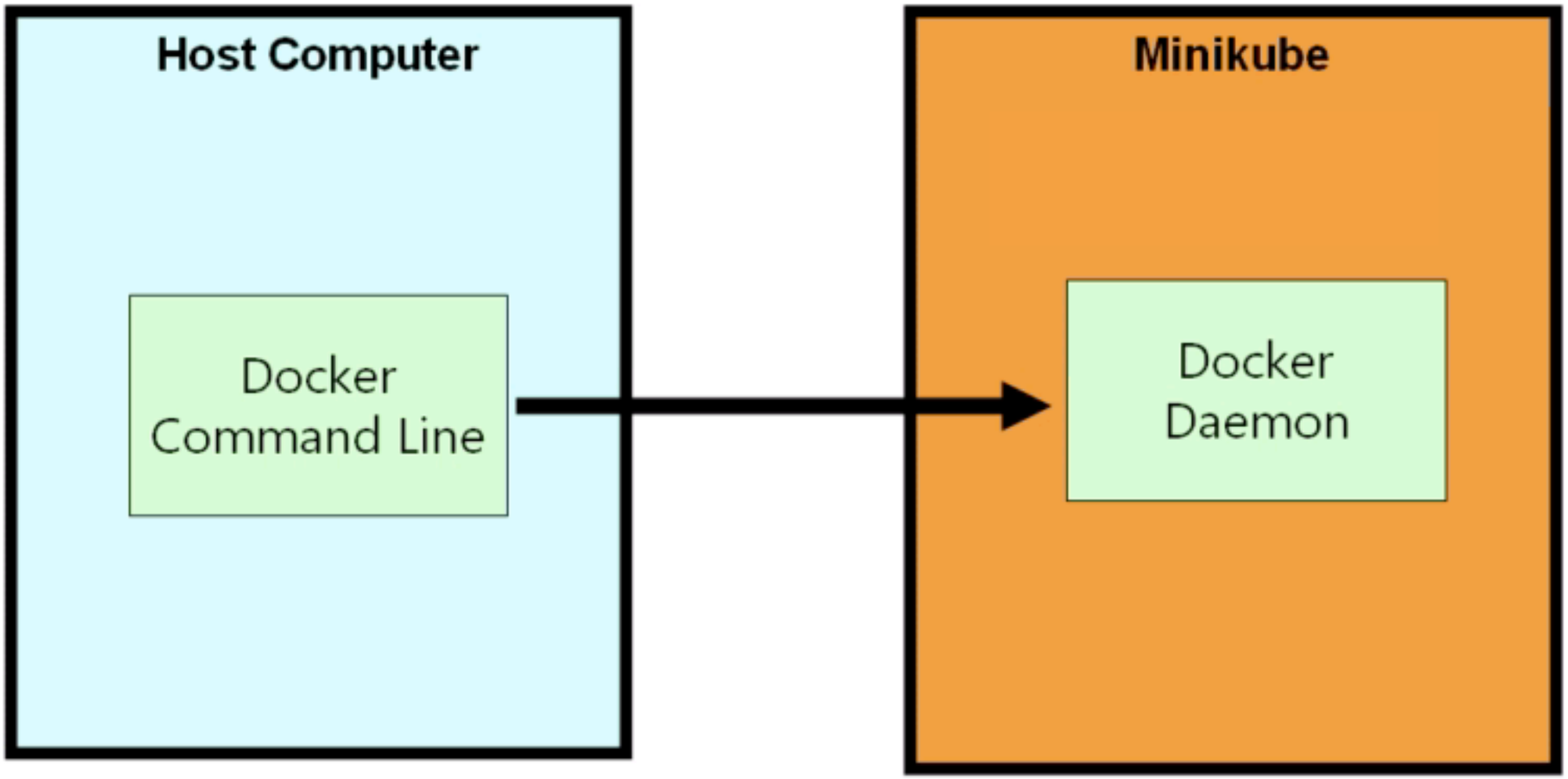
```
~                                                                    77  17:12:20
$ docker container run -p 8080:80 -d richardchesterwood/k8s-fleetman-webapp-angular:release0-5
cc4be1f5ab9d82be5f8be5c16a2ff0e73155c073852b413c2f9d740ecfc196c3
```

- 8080 - the port we want to expose to the outside world
- Internally the traffic will be routed to port 80

- Let's see if the container is running

```
~                                                                                                                    79  17:16:50
$ docker container ls
CONTAINER ID        IMAGE                                                  COMMAND                  CREATED
      STATUS              PORTS                               NAMES
cc4be1f5ab9d        richardchesterwood/k8s-fleetman-webapp-angular:release0-5   "nginx -g 'daemon of…"   33 seconds ag
o      Up 32 seconds       0.0.0.0:8080->80/tcp   dazzling_goldwasser
d1a552366f0d        25be029ad6d0                                           "nginx -g 'daemon of…"   6 hours ago
       Up 6 hours                                 k8s_webapp_webapp-858957ccc9-dvs68_default_32ed9521-bc9a-4b4e-8f10-8
1dac098d046_3
4f2eddd97539        25be029ad6d0                                           "nginx -g 'daemon of…"   6 hours ago
       Up 6 hours                                 k8s_webapp_webapp-858957ccc9-vz9bm_default_1a01e296-5226-4aa6-b7d6-5
883f263bccb_3
4e416a36e6d0        383867b75fd2                                           "docker-entrypoint.s…"   6 hours ago
```

- Let's try http://localhost:8080



- The Minikube docker daemon can't be accessible from our computer

Let's figure out the Minikube ip

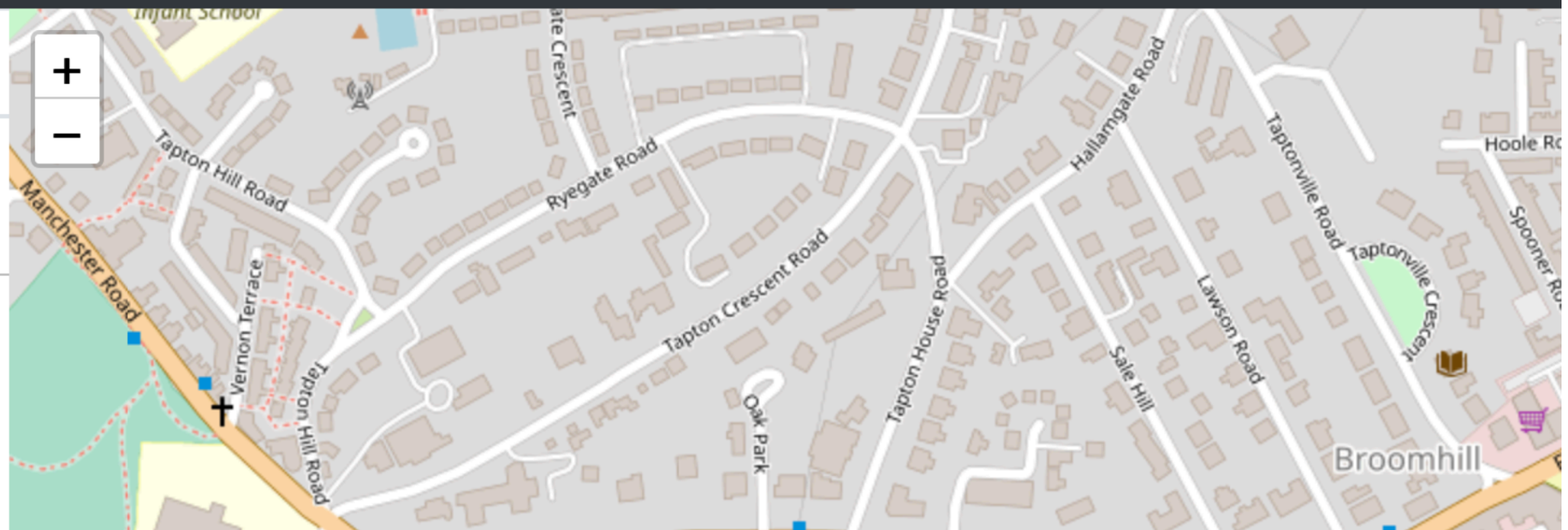
```
~
$ minikube ip
192.168.99.100
```

Let's try http://192.168.99.100:8080

Fleet Management System PROTOTYPE. Release 0.5

Name	Last seen	Speed mph

Live vehicle updates will appear here.
Once we've implemented it!



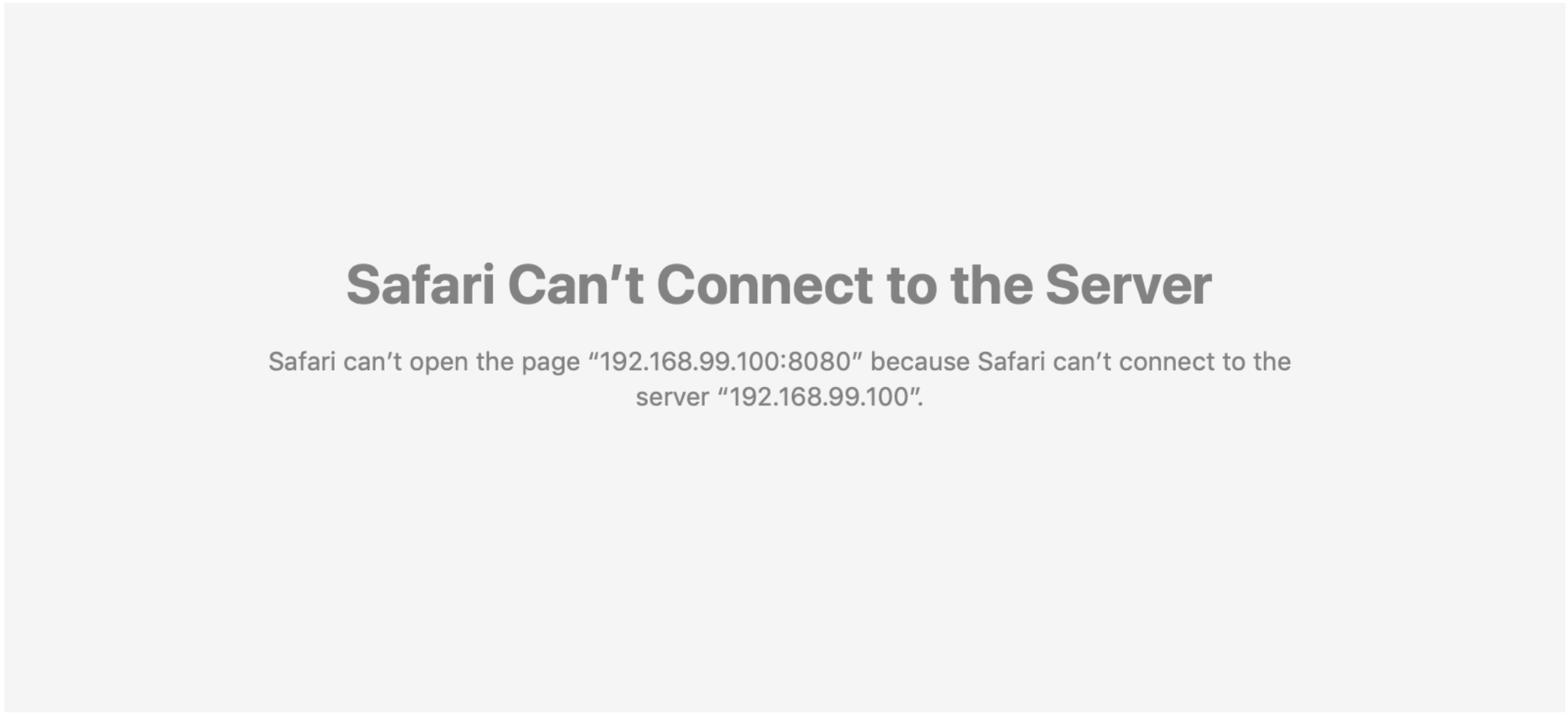
How to stop it?

```
~
$ docker container stop cc4
cc4
```

And remove it

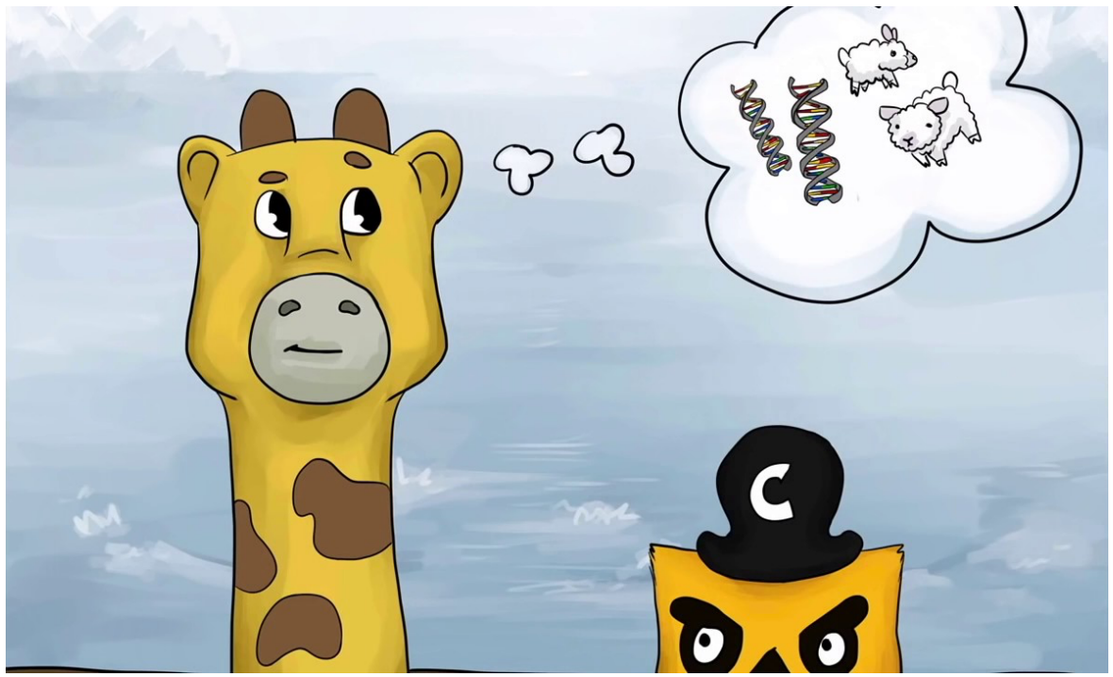
```
~
$ docker container rm cc4
cc4
```

- Let's try http://192.168.99.100:8080



P.PORTO

- https://martinfowler.com/articles/microservices.html
- https://kubernetes.io/docs/tutorials/kubernetes-basics/
- http://microservices.io



- https://www.youtube.com/watch?v=Q4W8Z-D-gcQ

P.PORTO

Polytechnic of
Porto

ESTG - School of
Management and Technology