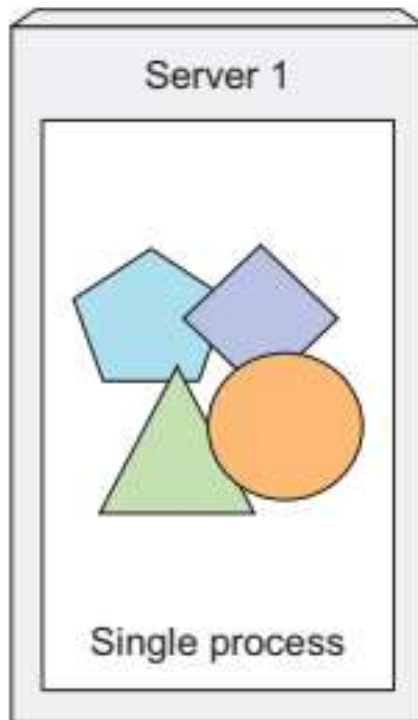


Kubernetes

Monolithic application



Microservices-based application

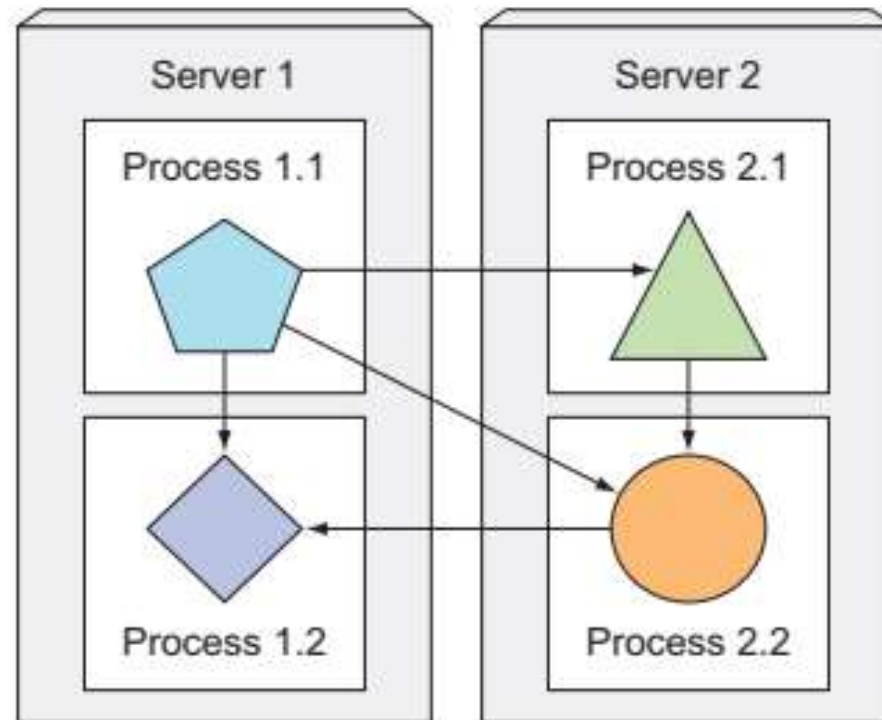
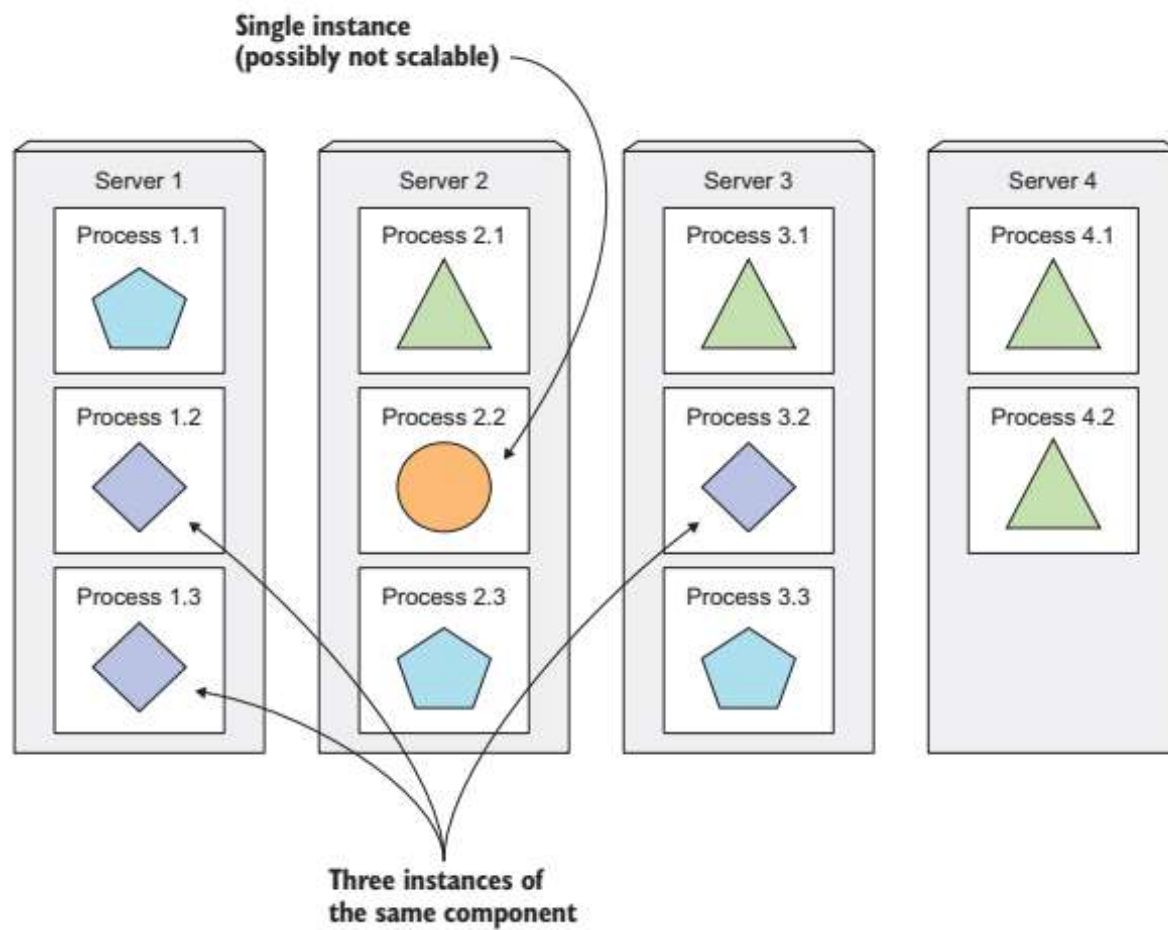
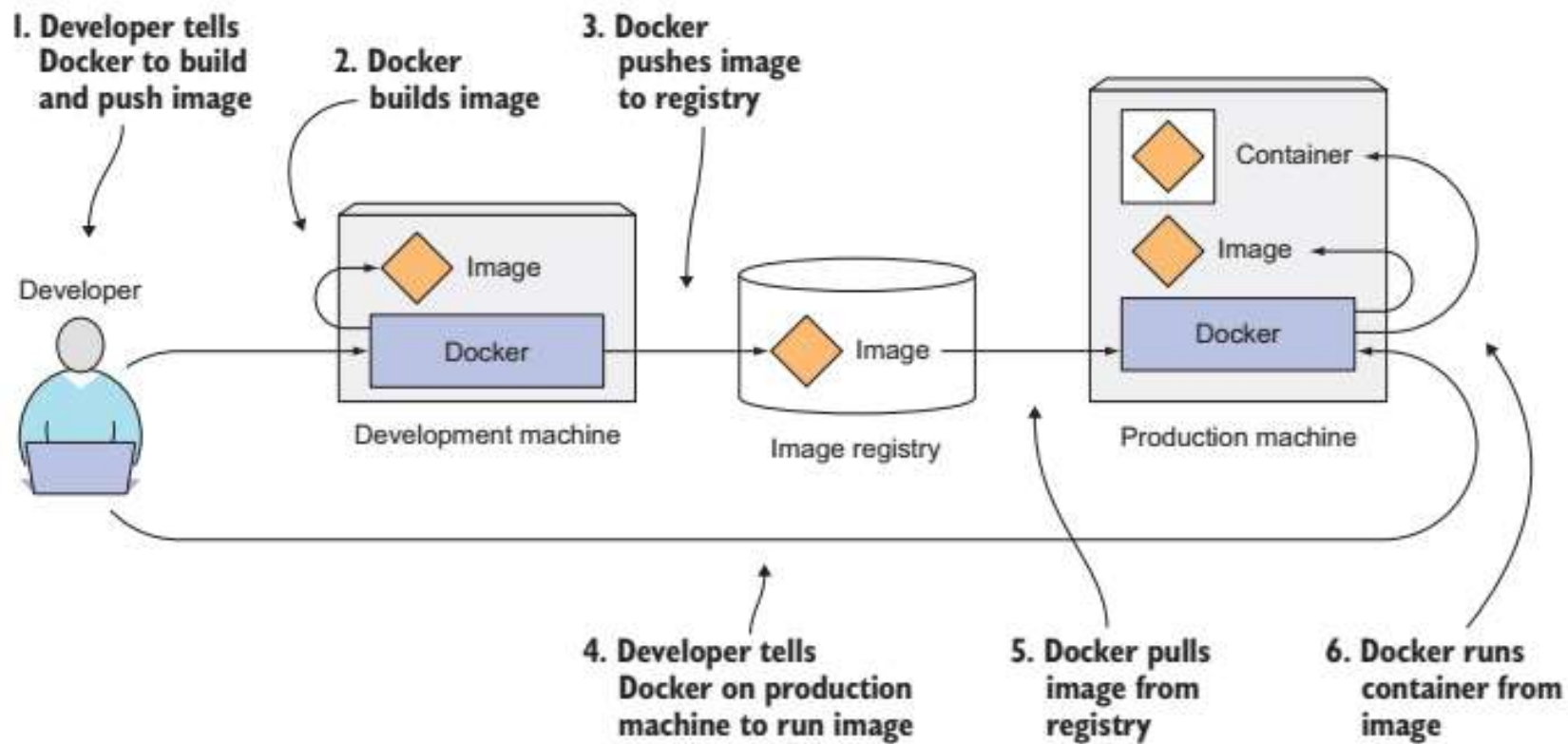
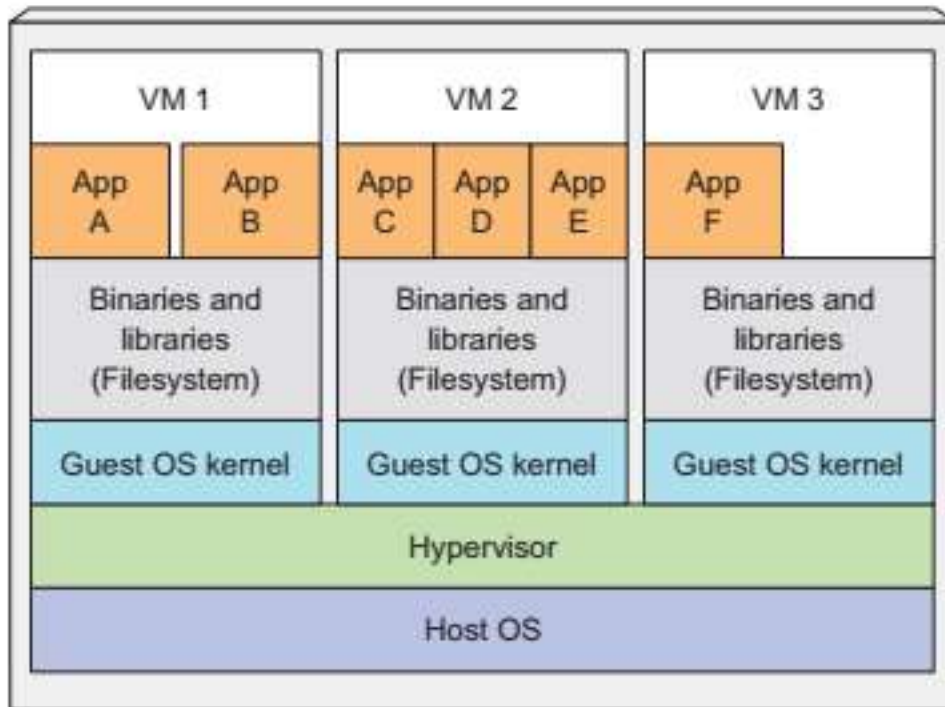


Figure 1.1 Components inside a monolithic application vs. standalone microservices

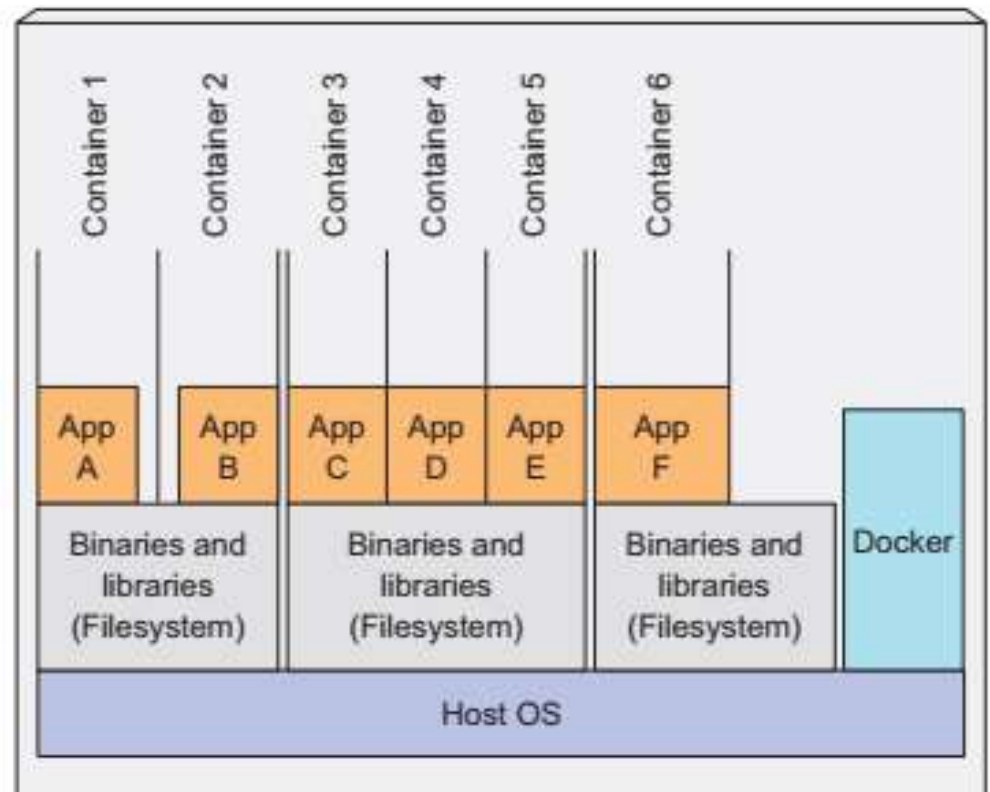




Host running multiple VMs



Host running multiple Docker containers



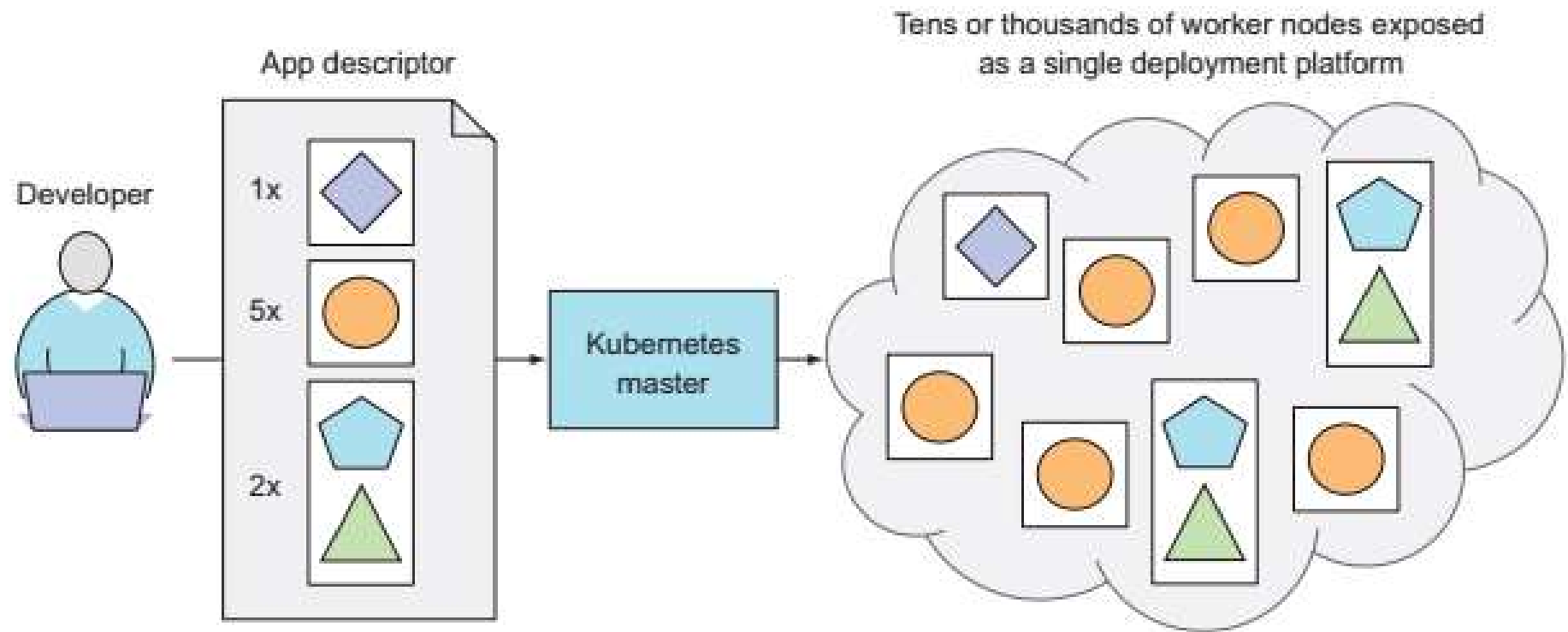


Figure 1.8 Kubernetes exposes the whole datacenter as a single deployment platform.

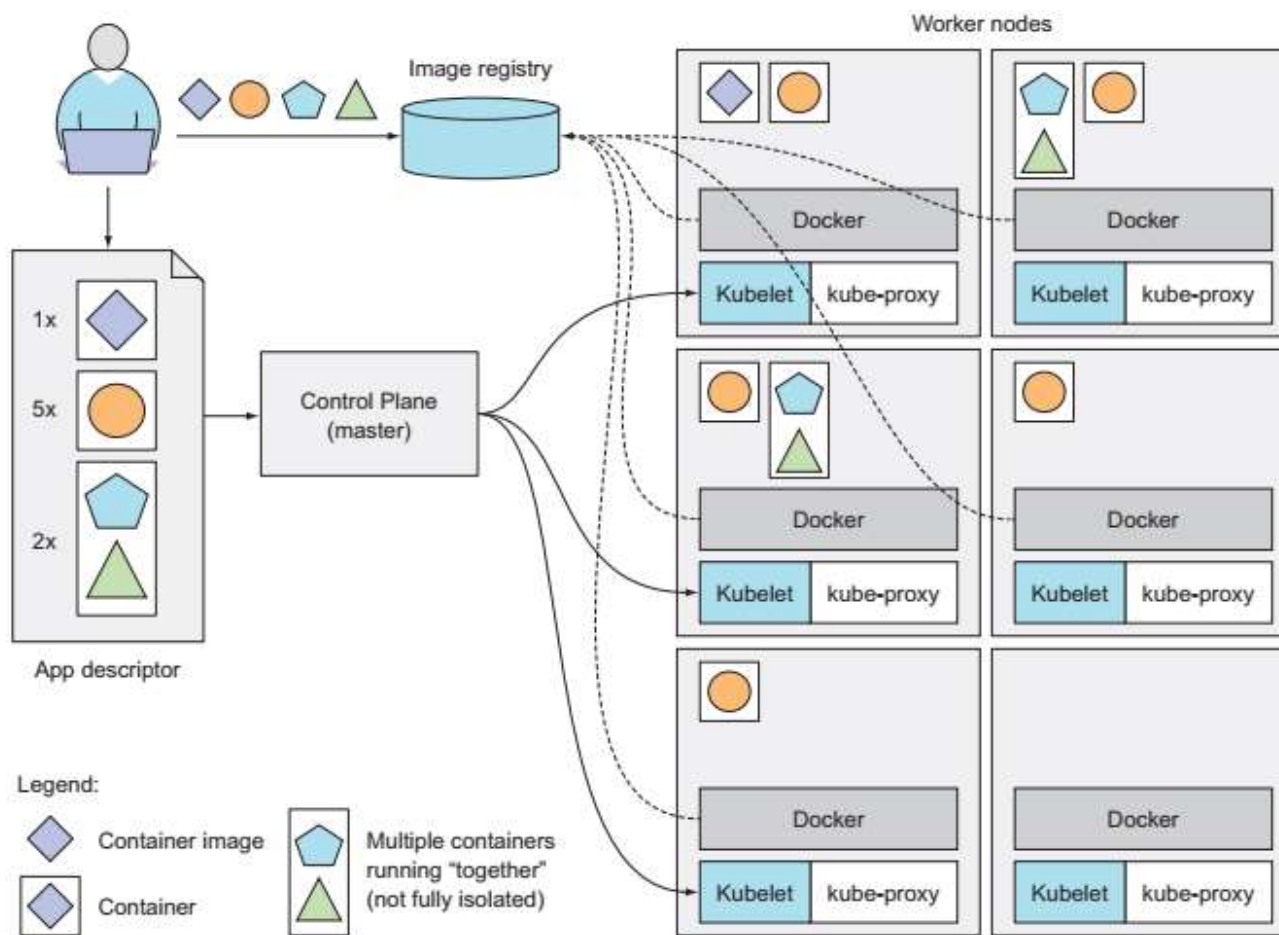


Figure 1.10 A basic overview of the Kubernetes architecture and an application running on top of it

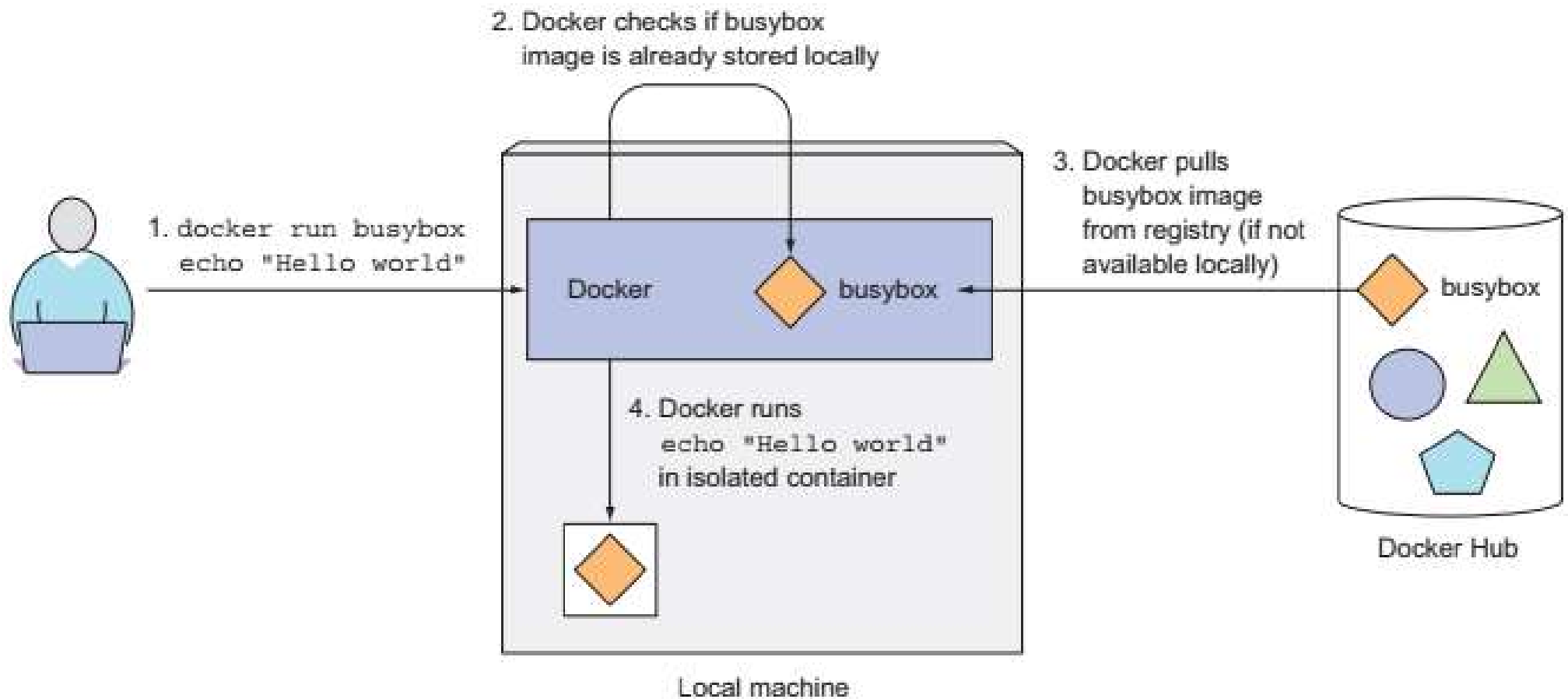


Figure 2.1 Running echo "Hello world" in a container based on the busybox container image

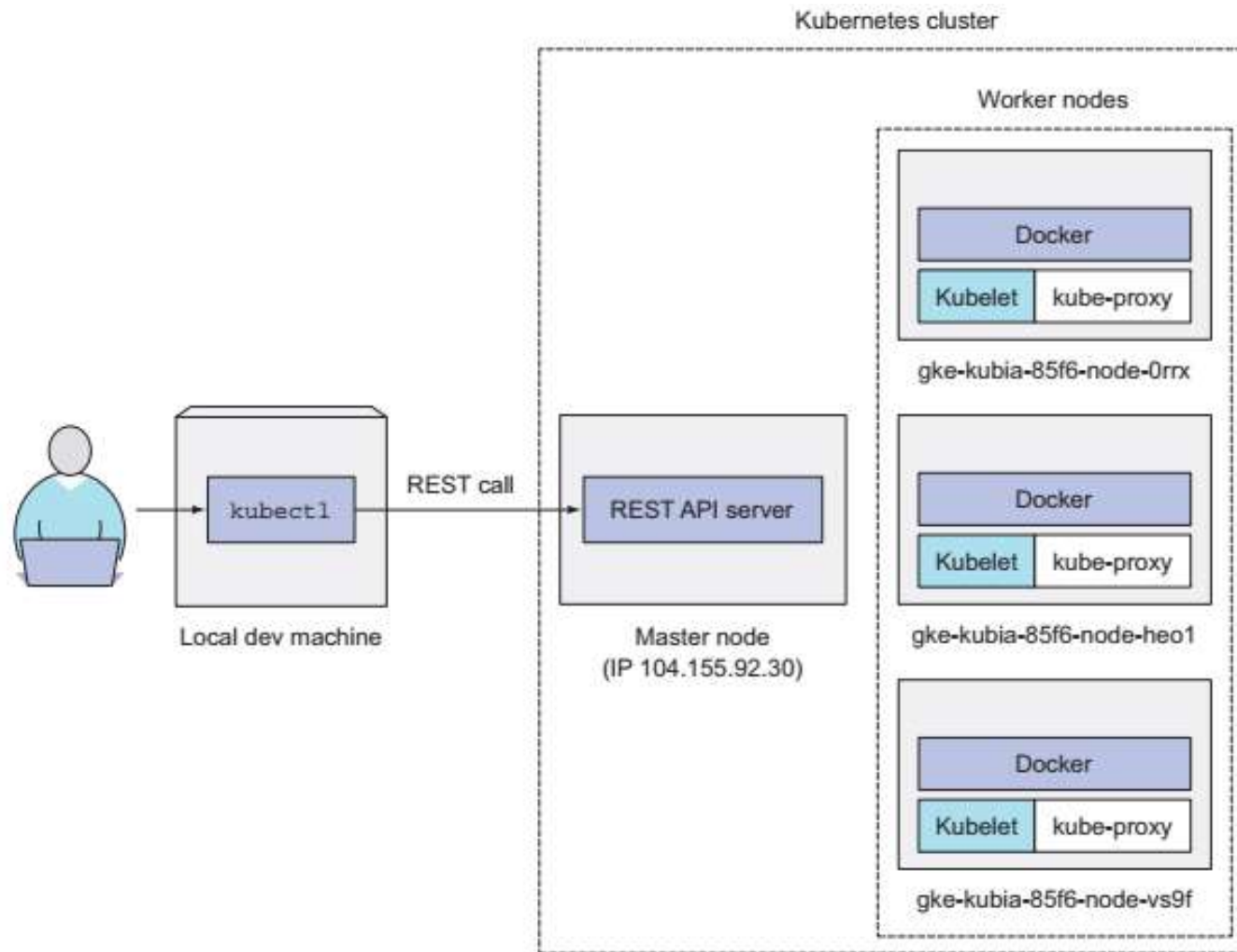


Figure 2.4 How you're interacting with your three-node Kubernetes cluster

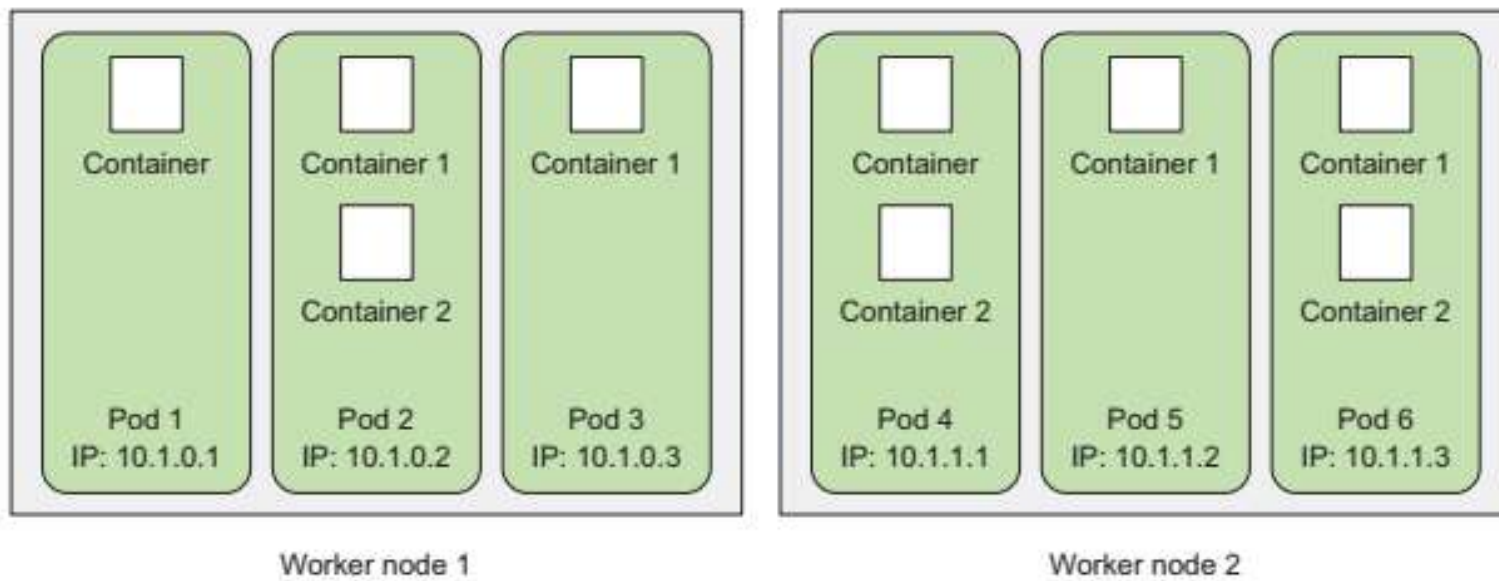


Figure 2.5 The relationship between containers, pods, and physical worker nodes

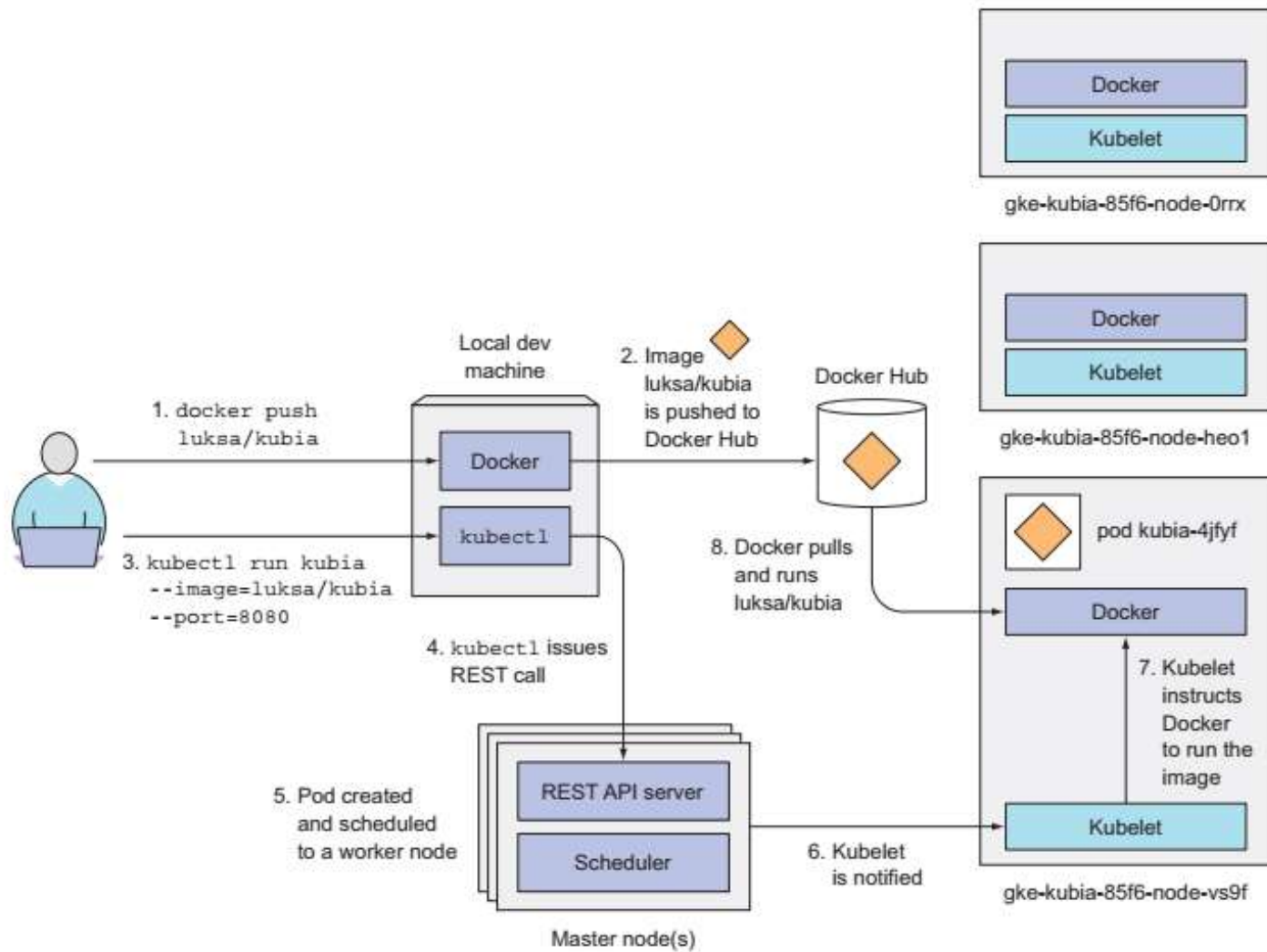


Figure 2.6 Running the luksa/kubia container image in Kubernetes

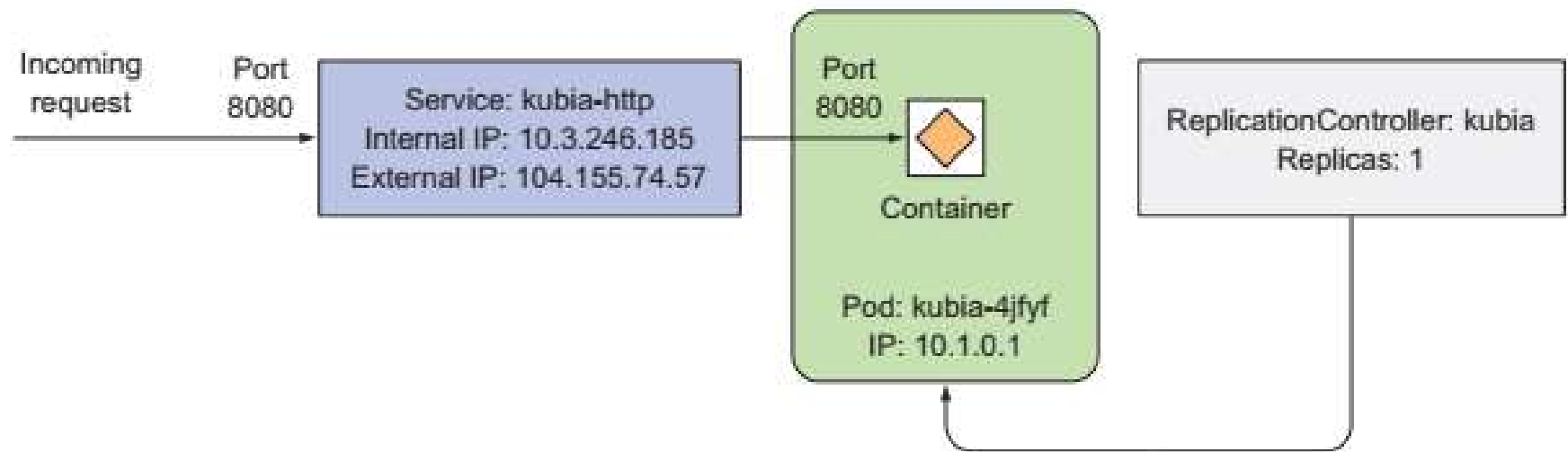


Figure 2.7 Your system consists of a ReplicationController, a Pod, and a Service.

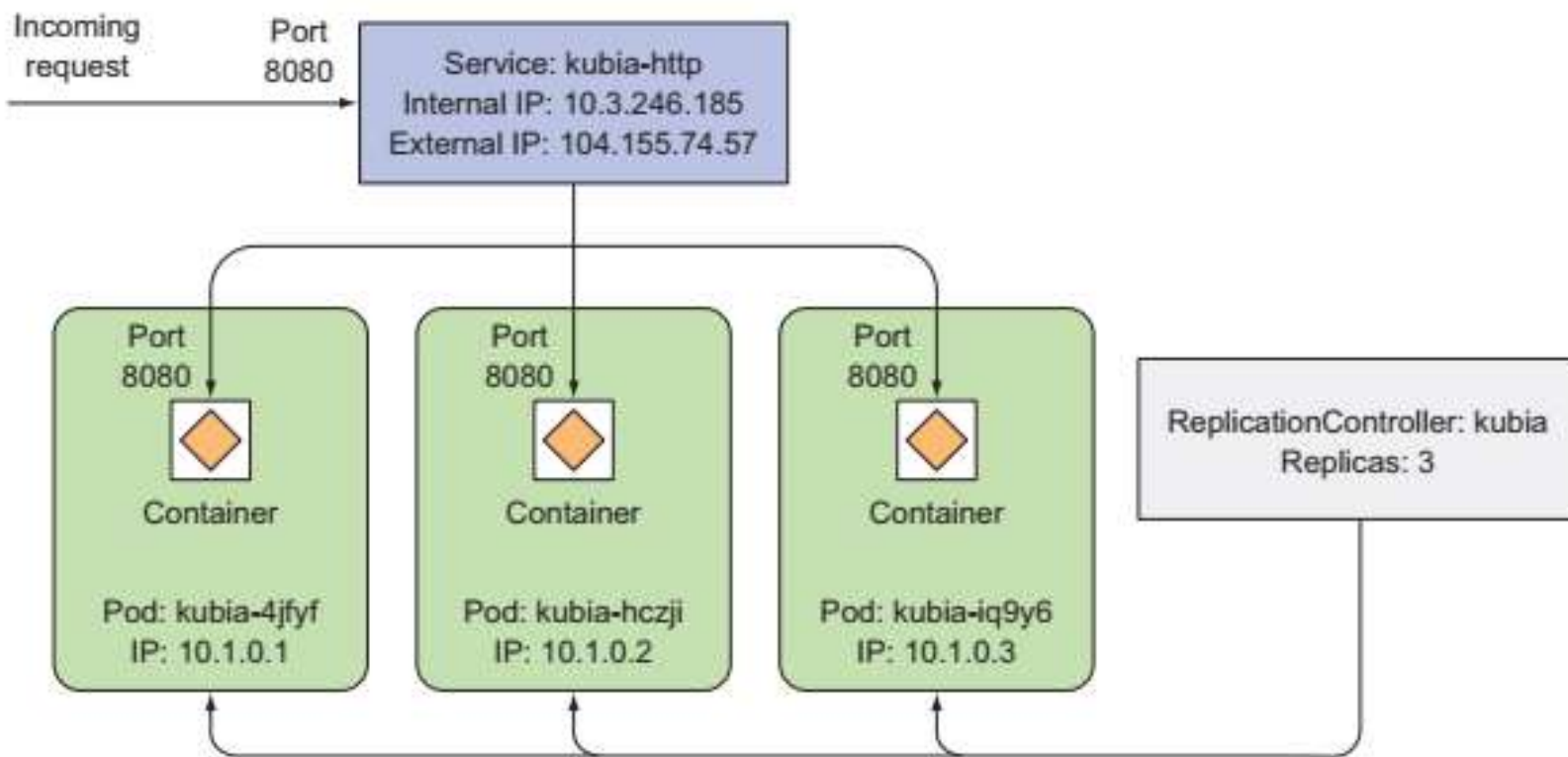


Figure 2.8 Three instances of a pod managed by the same ReplicationController and exposed through a single service IP and port.

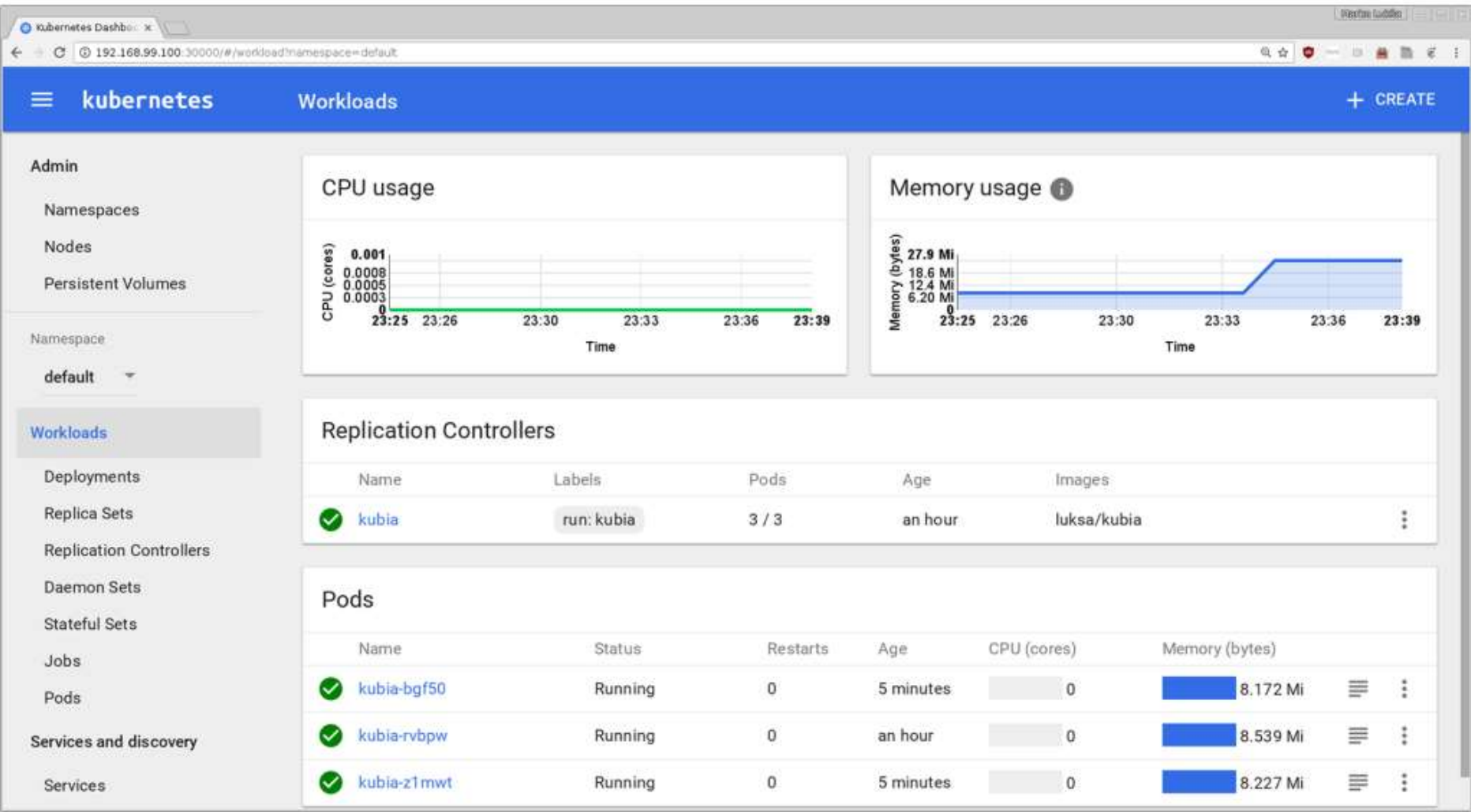


Figure 2.9 Screenshot of the Kubernetes web-based dashboard

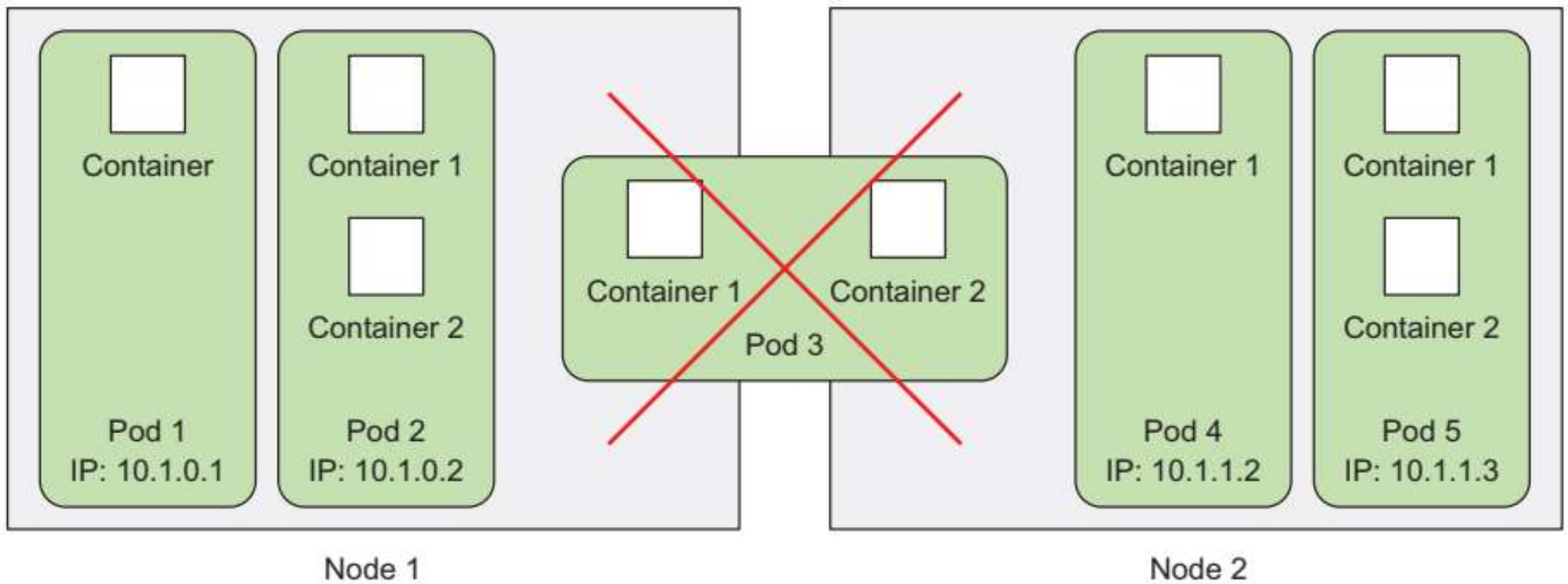


Figure 3.1 All containers of a pod run on the same node. A pod never spans two nodes.

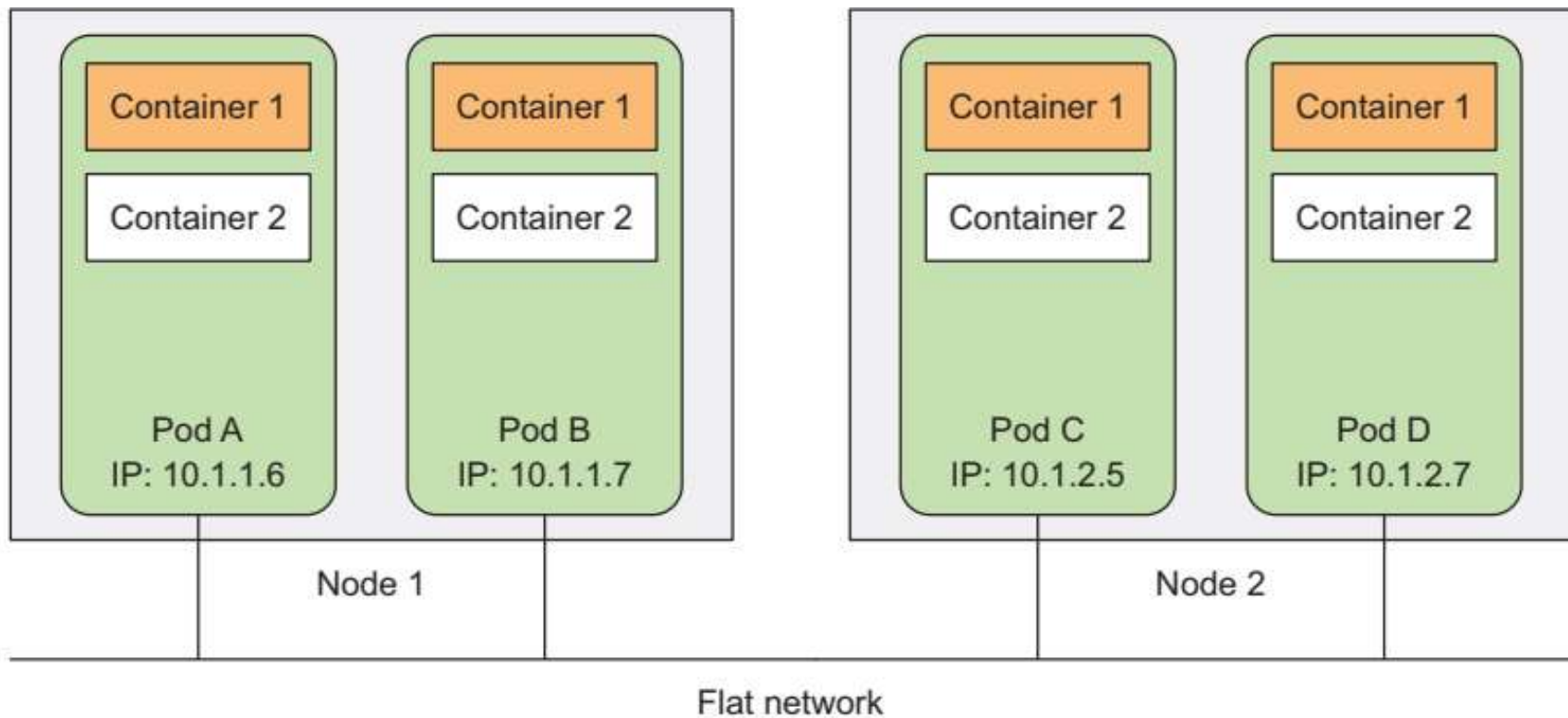


Figure 3.2 Each pod gets a routable IP address and all other pods see the pod under that IP address.

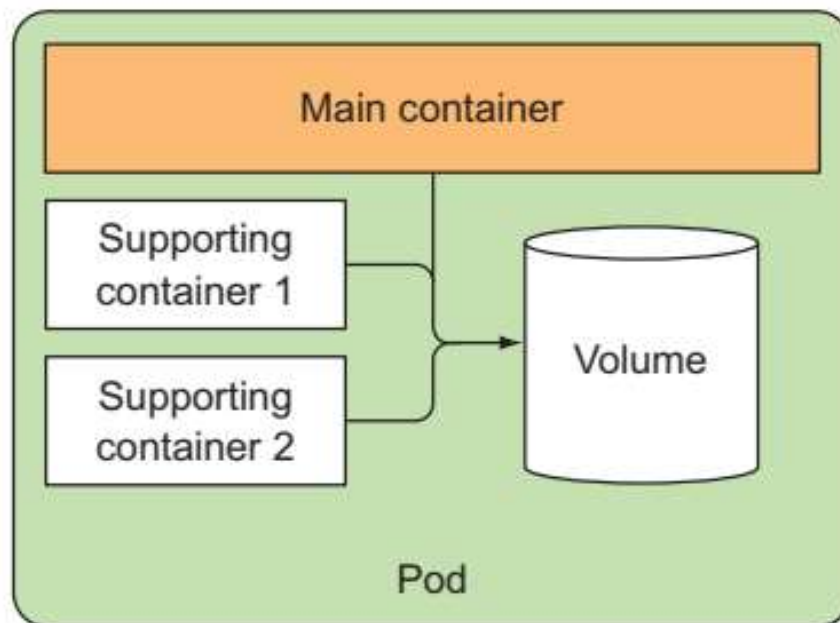
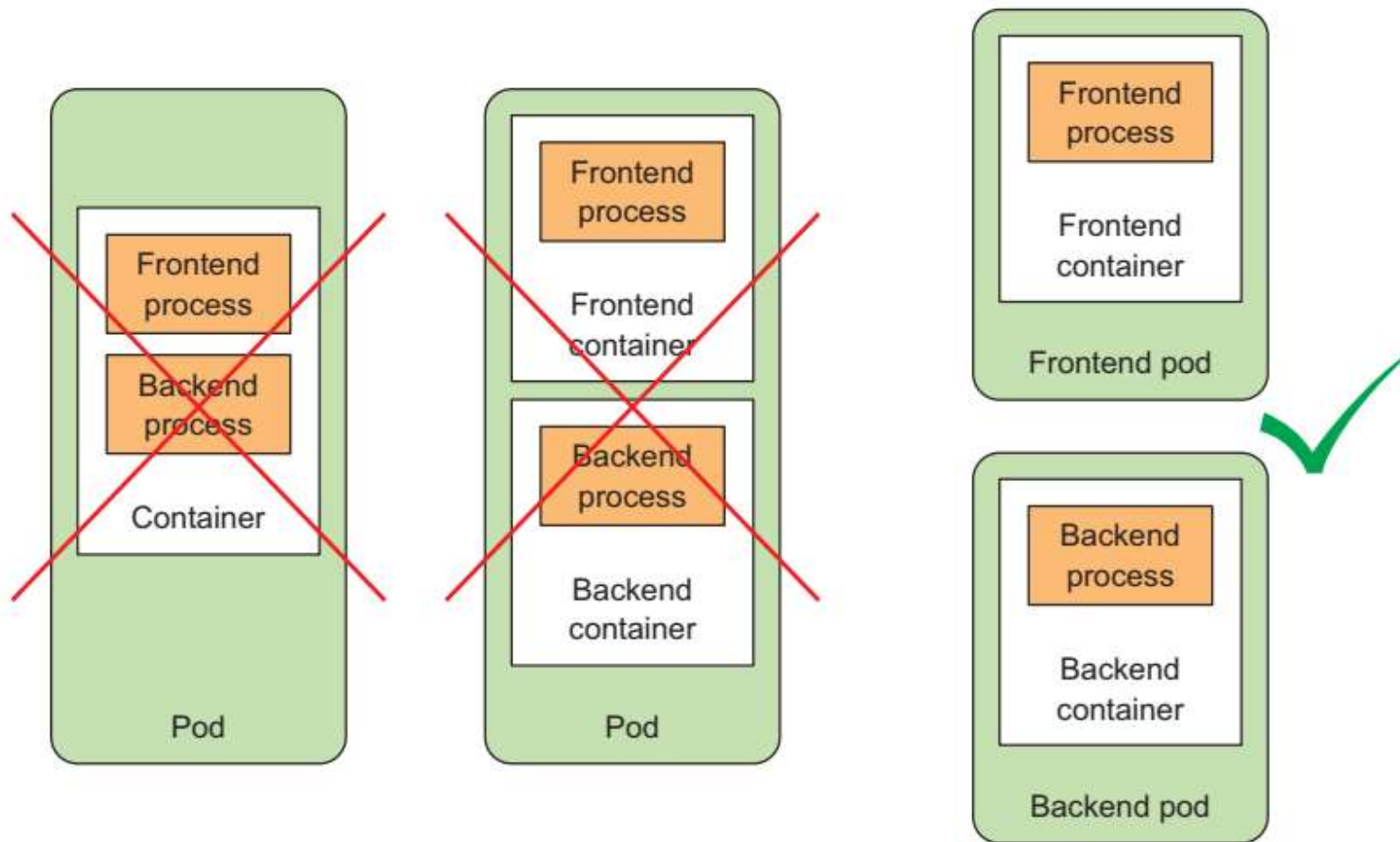
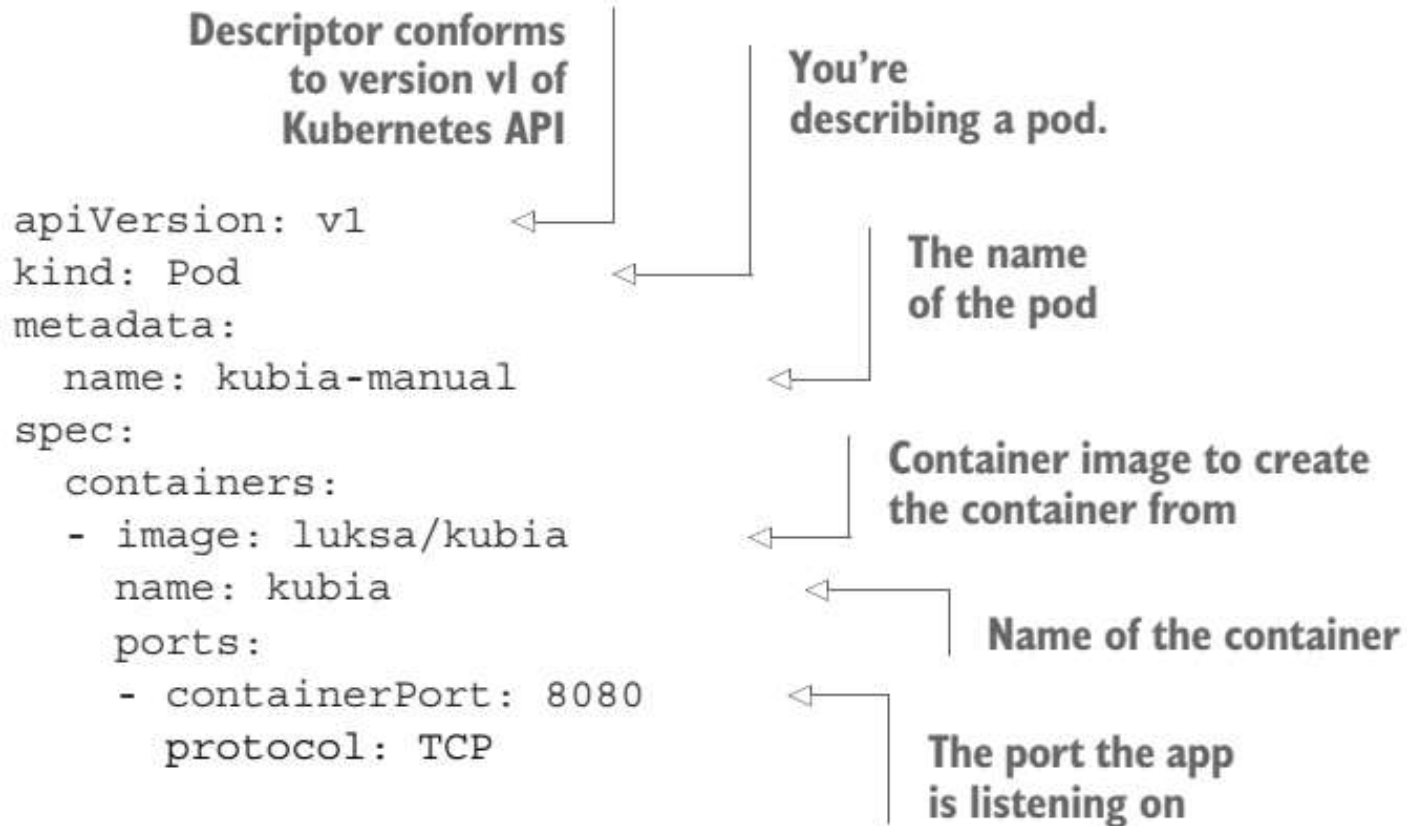


Figure 3.3 Pods should contain tightly coupled containers, usually a main container and containers that support the main one.



Listing 3.2 A basic pod manifest: kuba-manual.yaml



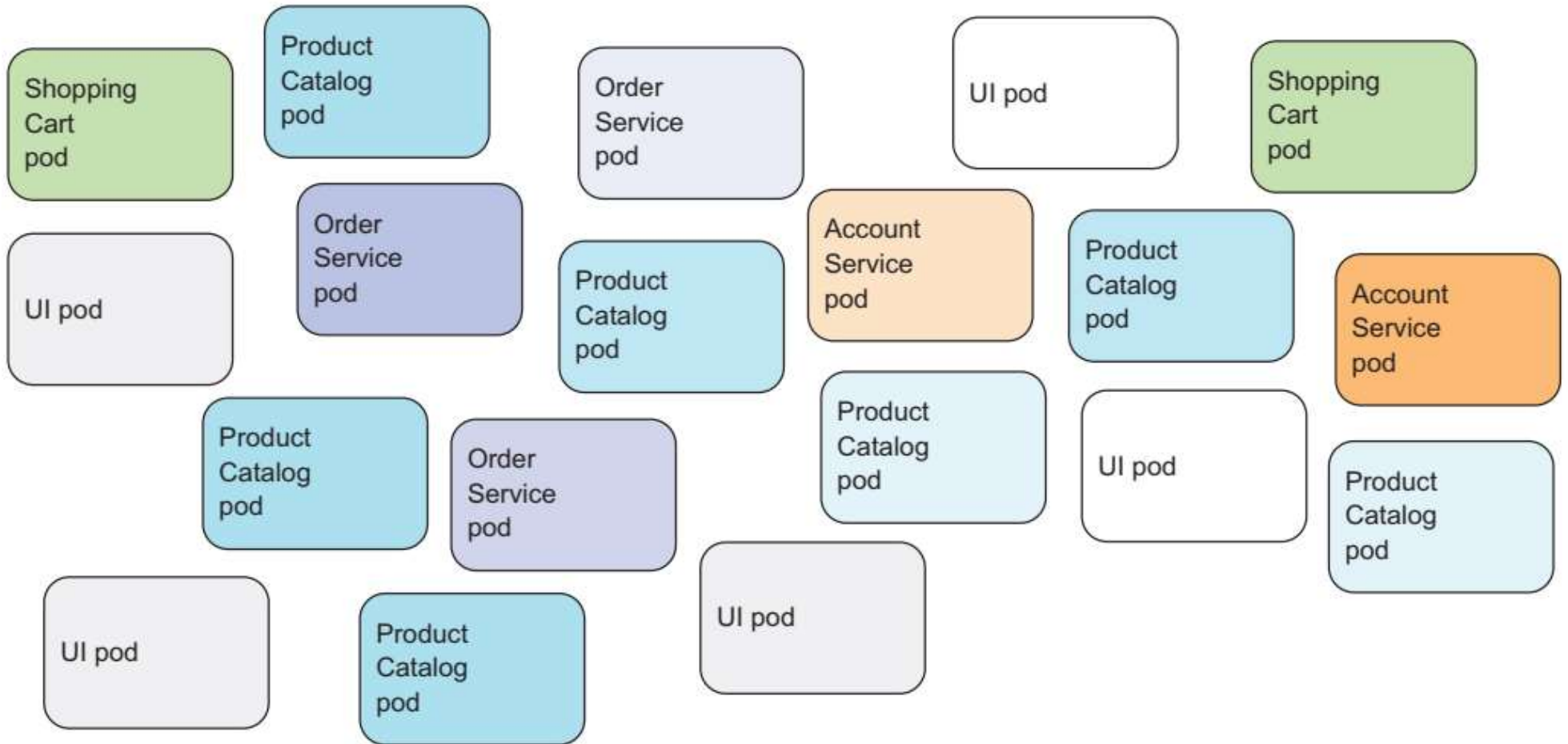


Figure 3.6 Uncategorized pods in a microservices architecture

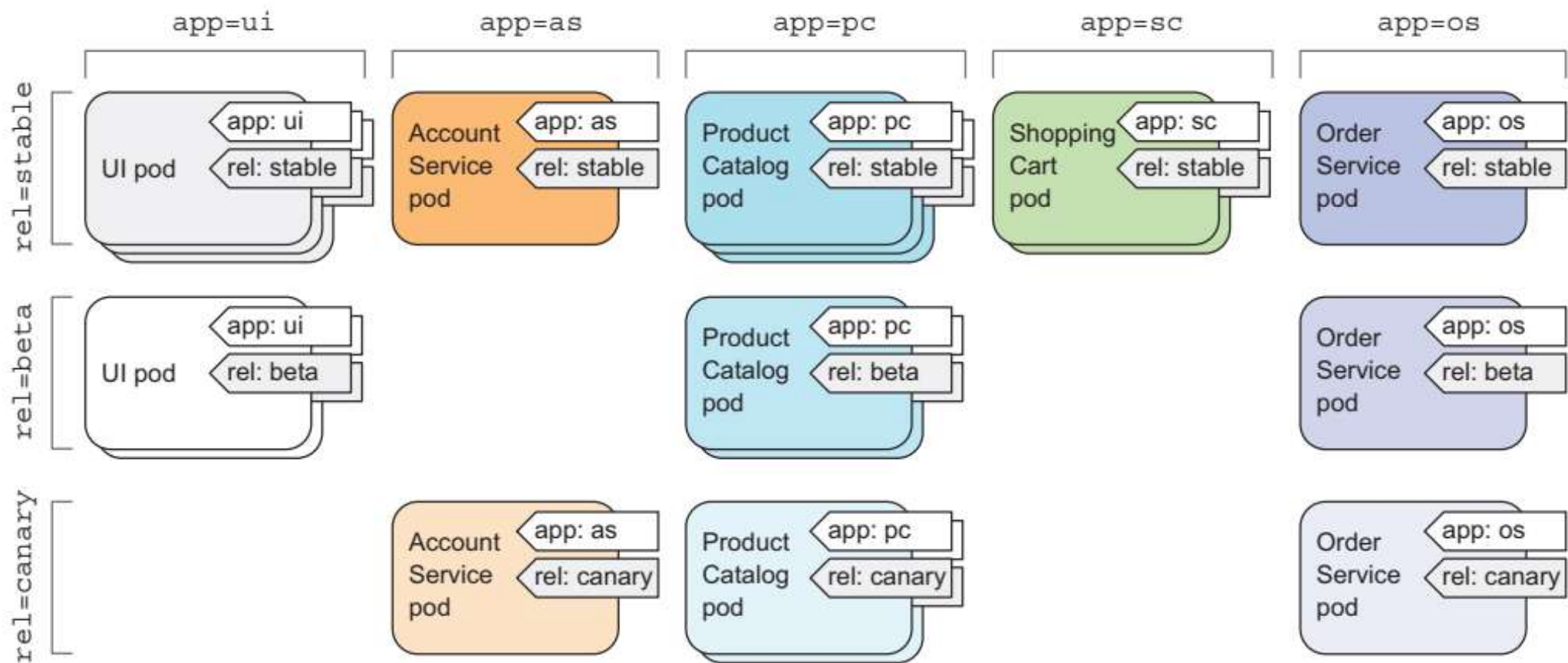


Figure 3.7 Organizing pods in a microservices architecture with pod labels

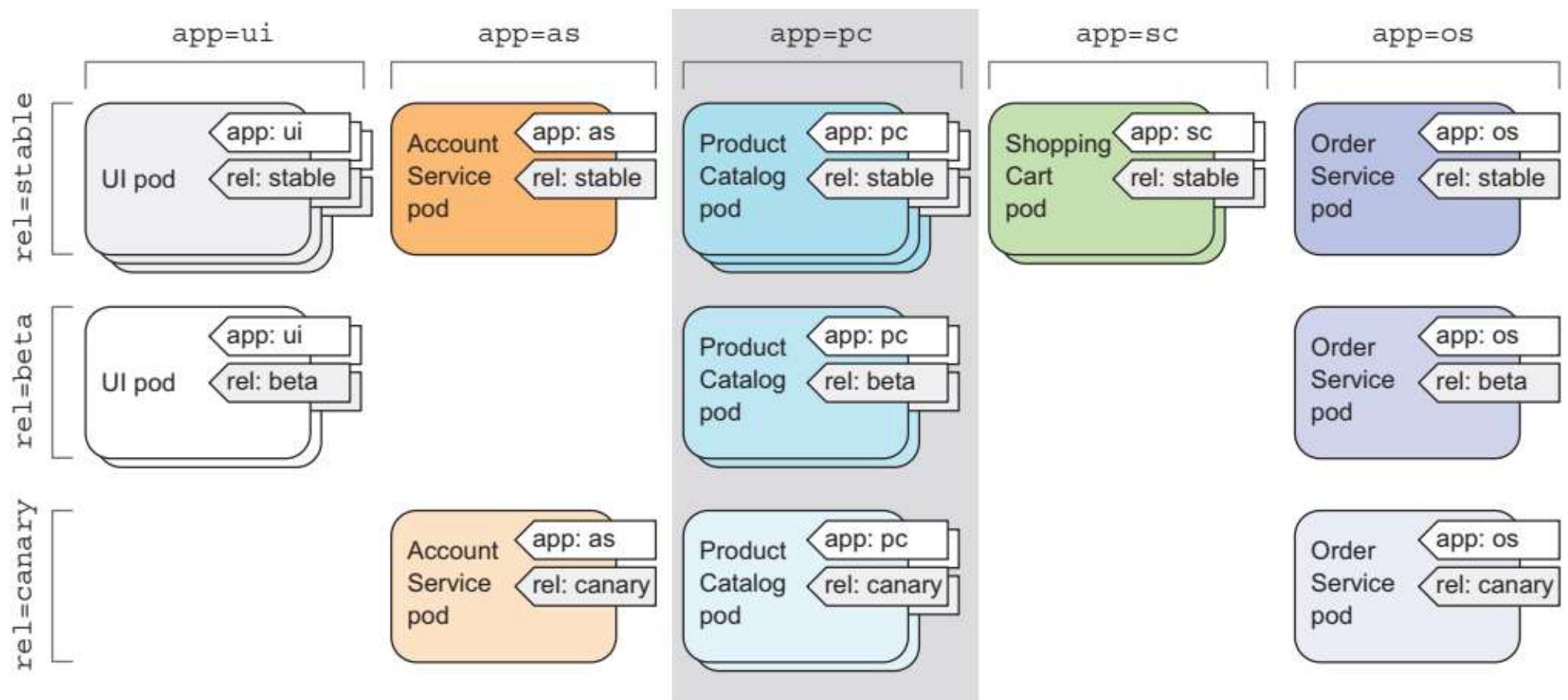


Figure 3.8 Selecting the product catalog microservice pods using the “app=pc” label selector

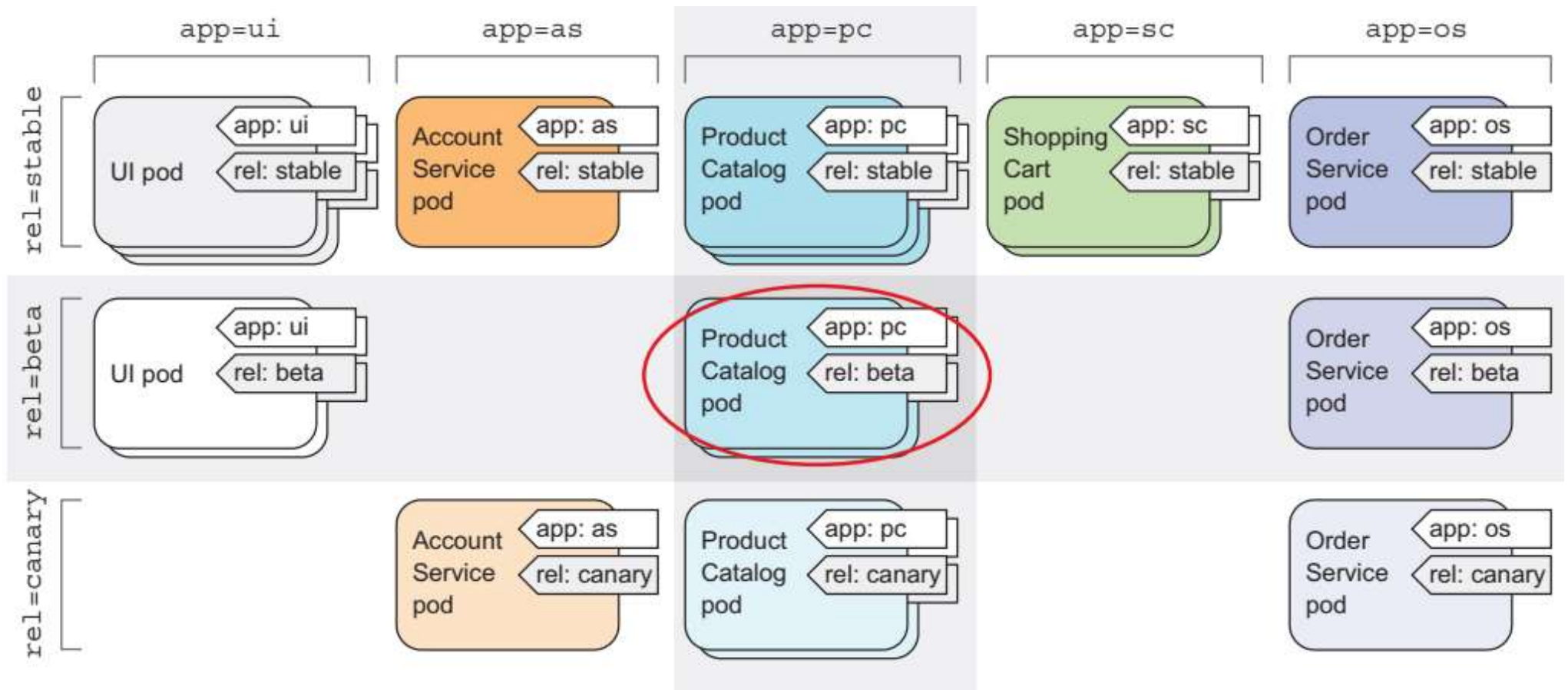


Figure 3.9 Selecting pods with multiple label selectors

Stopping and removing pods

81

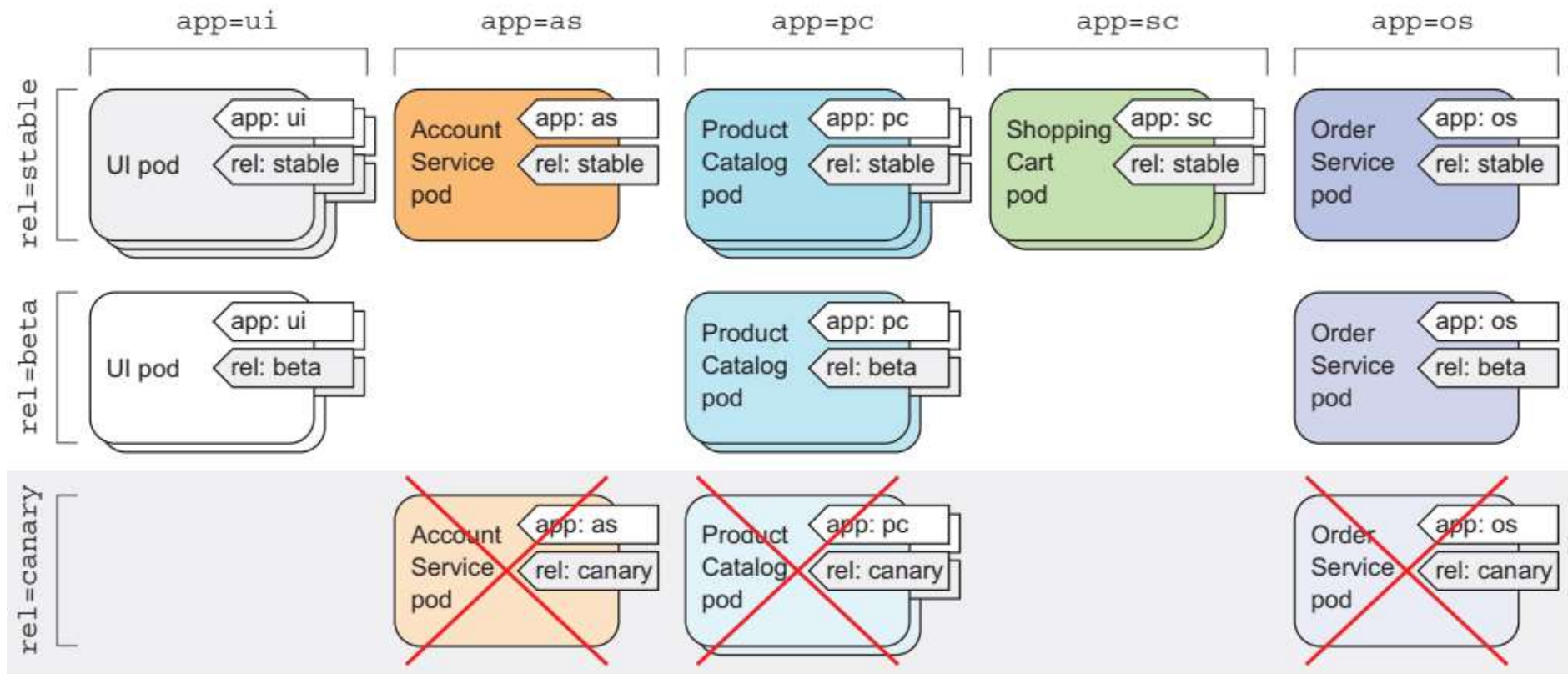


Figure 3.10 Selecting and deleting all canary pods through the `rel=canary` label selector

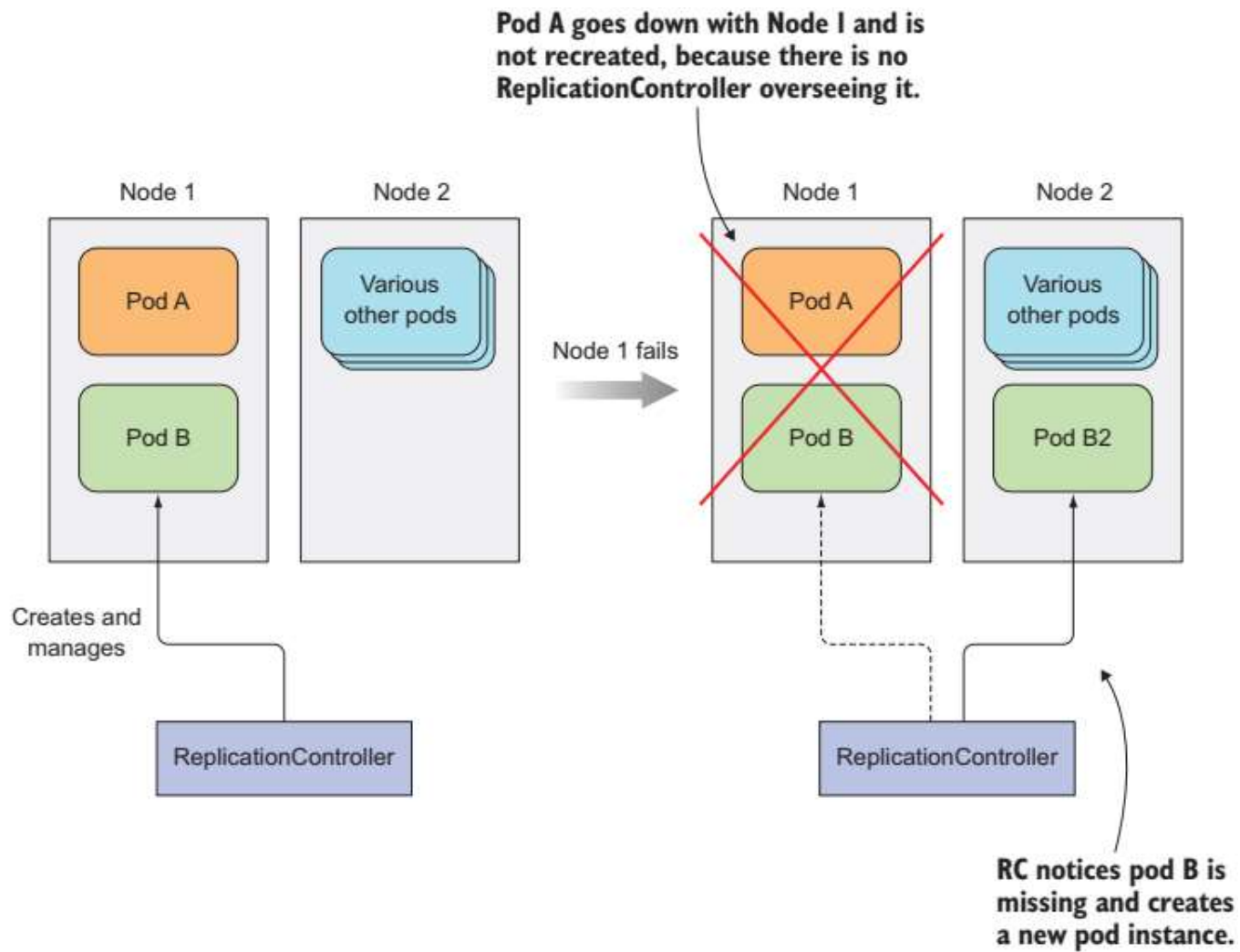


Figure 4.1 When a node fails, only pods backed by a ReplicationController are recreated.

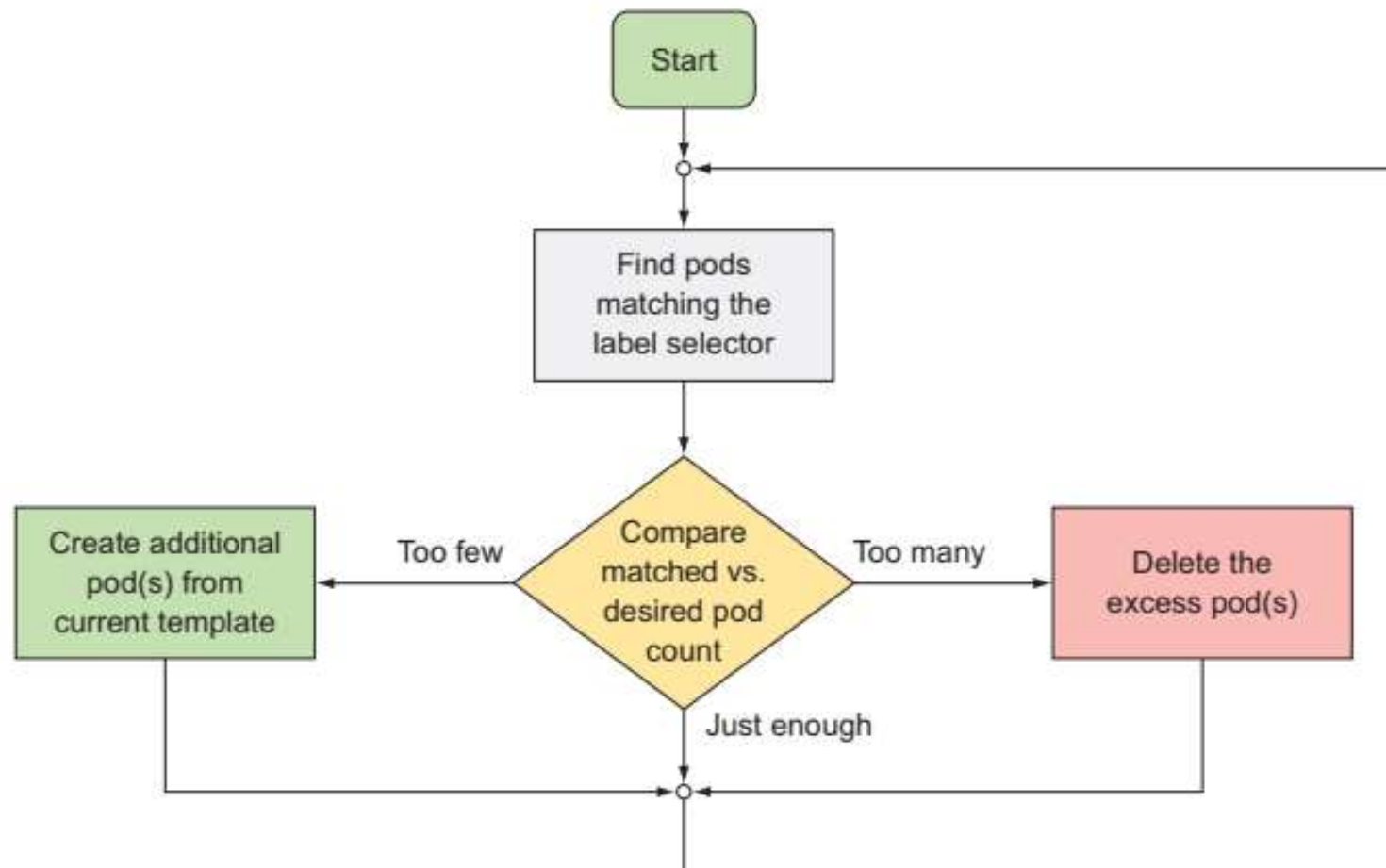


Figure 4.2 A ReplicationController's reconciliation loop

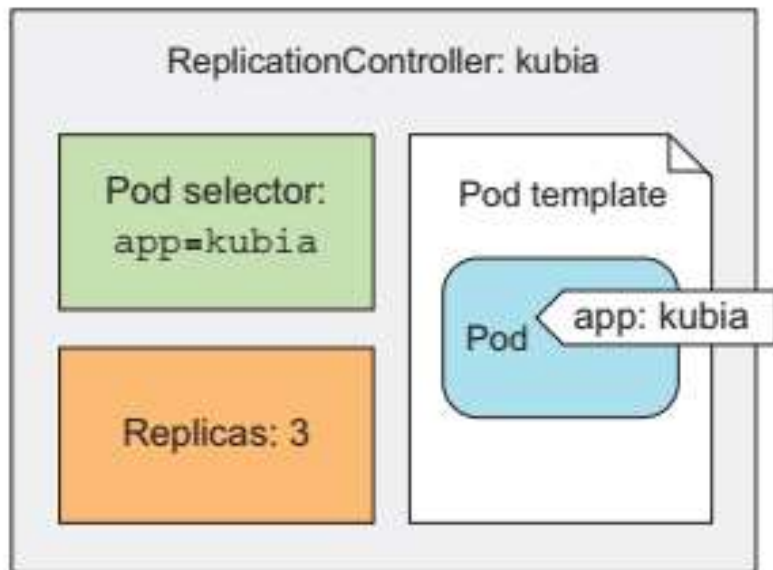


Figure 4.3 The three key parts of a `ReplicationController` (pod selector, replica count, and pod template)

Listing 4.4 A YAML definition of a ReplicationController: kubia-rc.yaml

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: kubia
spec:
  replicas: 3
  selector:
    app: kubia
```

This manifest defines a
ReplicationController (RC)

The name of this
ReplicationController

The desired number
of pod instances

The pod selector determining
what pods the RC is operating on

```
template:
  metadata:
    labels:
      app: kubia
  spec:
    containers:
      - name: kubia
        image: luksa/kubia
        ports:
          - containerPort: 8080
```

The pod template
for creating new
pods

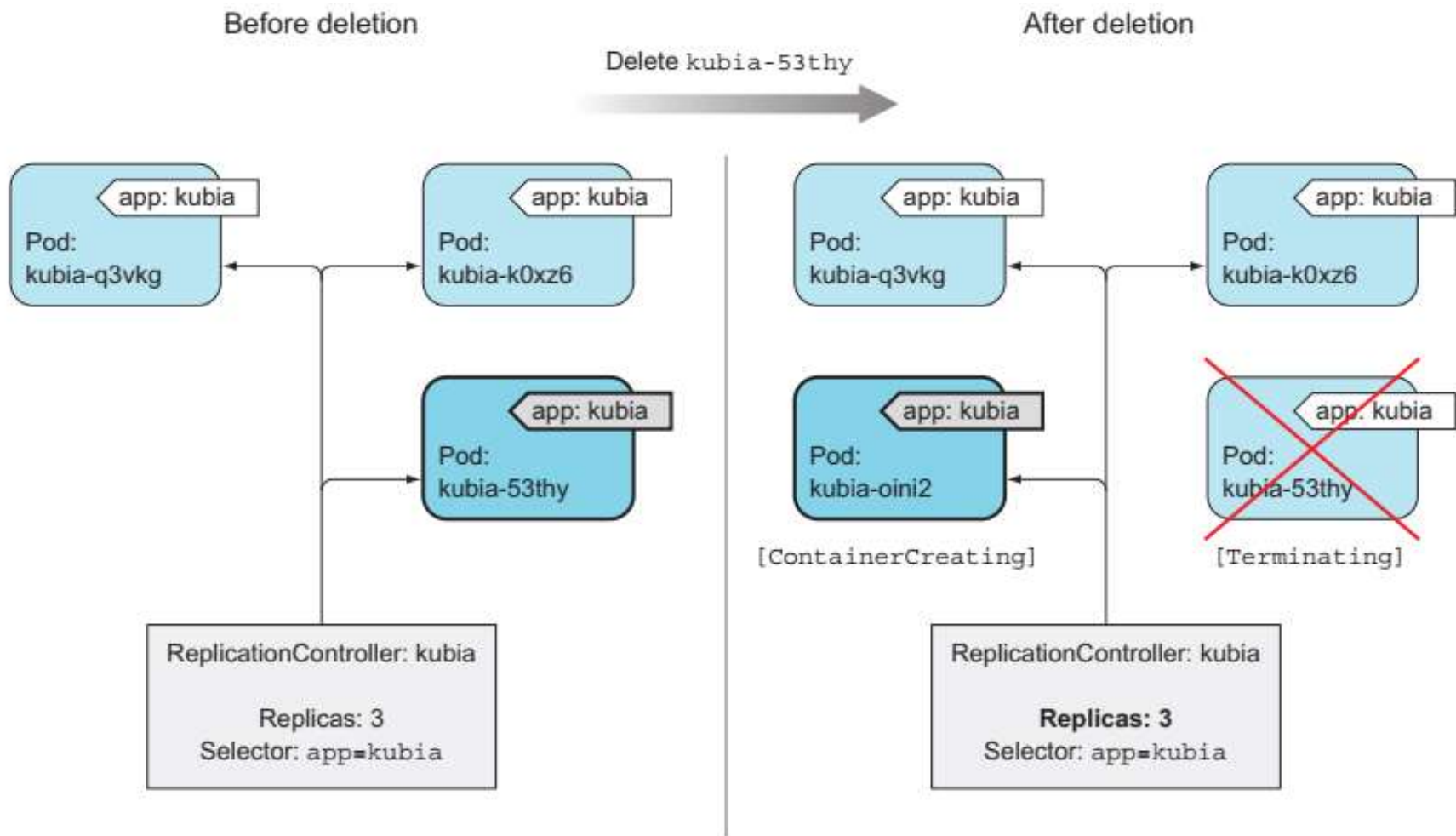


Figure 4.4 If a pod disappears, the ReplicationController sees too few pods and creates a new replacement pod.

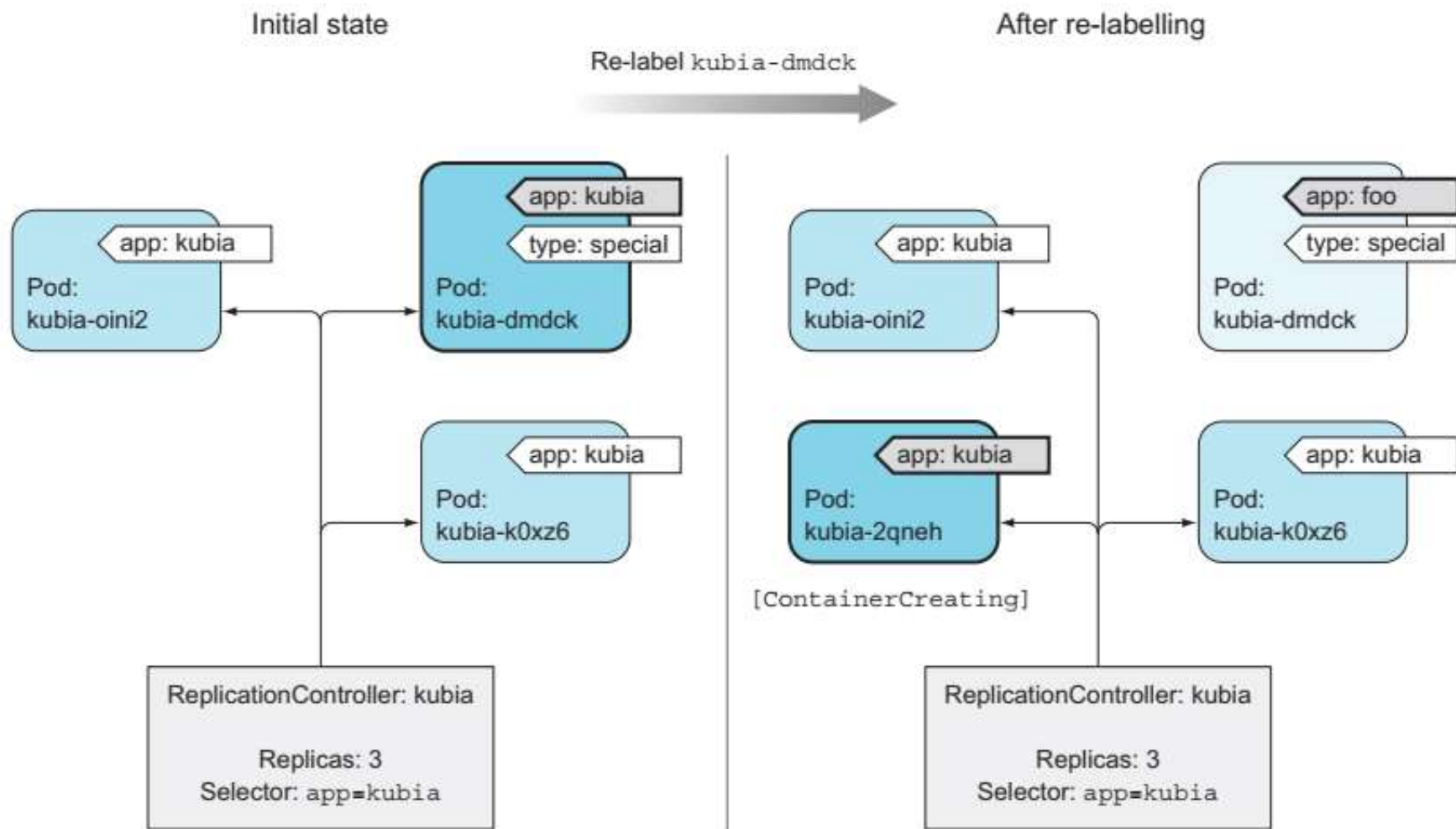


Figure 4.5 Removing a pod from the scope of a ReplicationController by changing its labels

Services

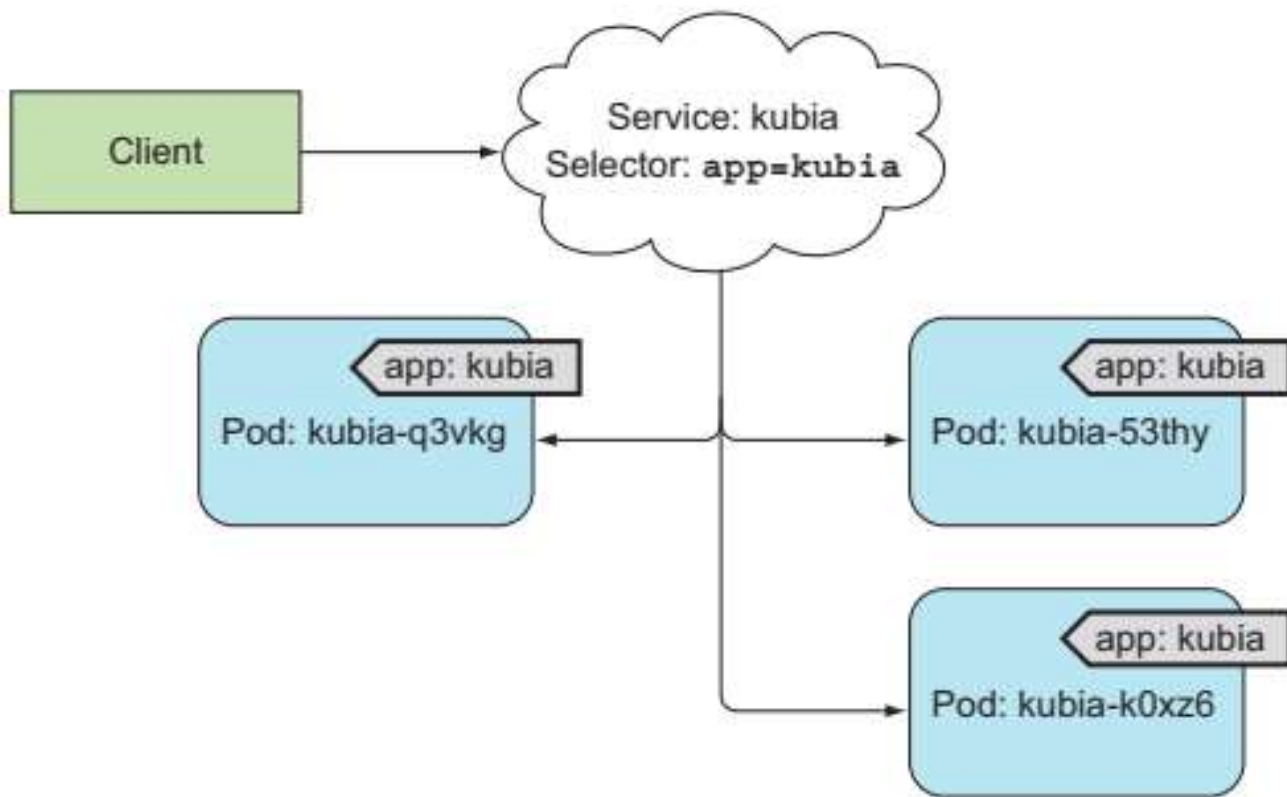


Figure 5.2 Label selectors determine which pods belong to the Service.

Listing 5.1 A definition of a service: kubia-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  ports:
    - port: 80
      targetPort: 8080
  selector:
    app: kubia
```

The port this service
will be available on

The container port the
service will forward to

All pods with the app=kubia
label will be part of this service.

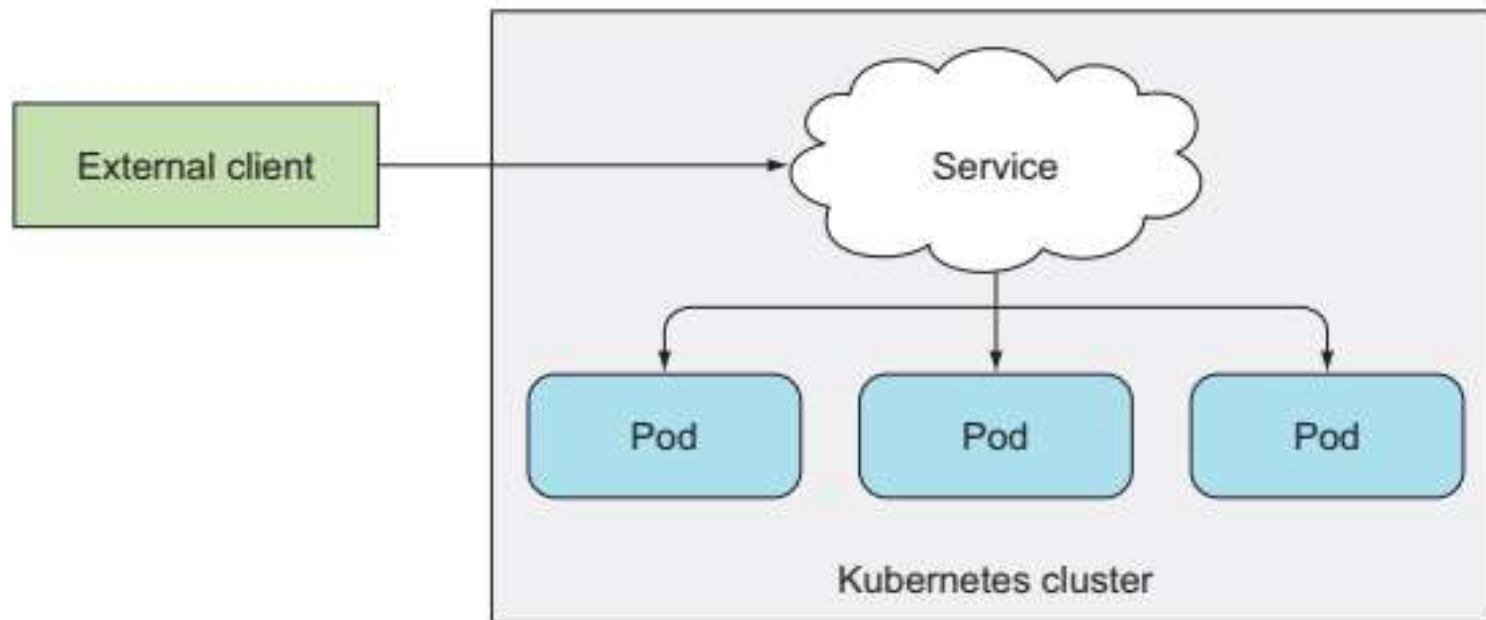
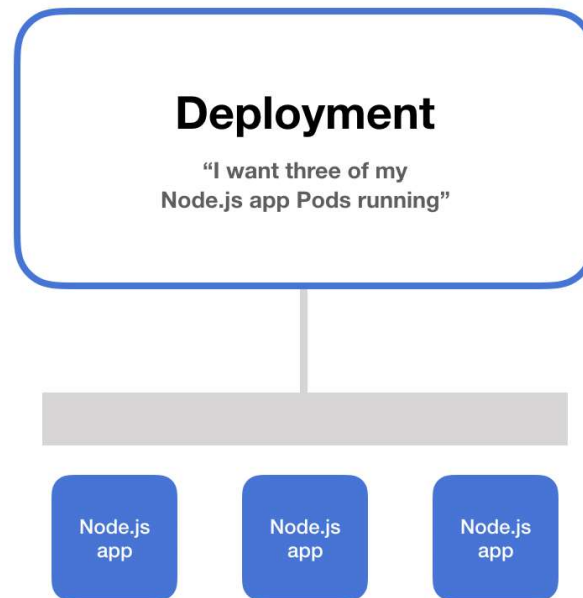


Figure 5.5 Exposing a service to external clients

Kubernetes deployment

- Everyone running applications on Kubernetes cluster uses a deployment.
- It's what you use to scale, roll out, and roll back versions of your applications.
- With a deployment, you tell Kubernetes how many copies of a Pod you want running. The deployment takes care of everything else.



Installation - Online

- <https://labs.play-with-k8s.com>

Thanks