Introduction to Kubernetes

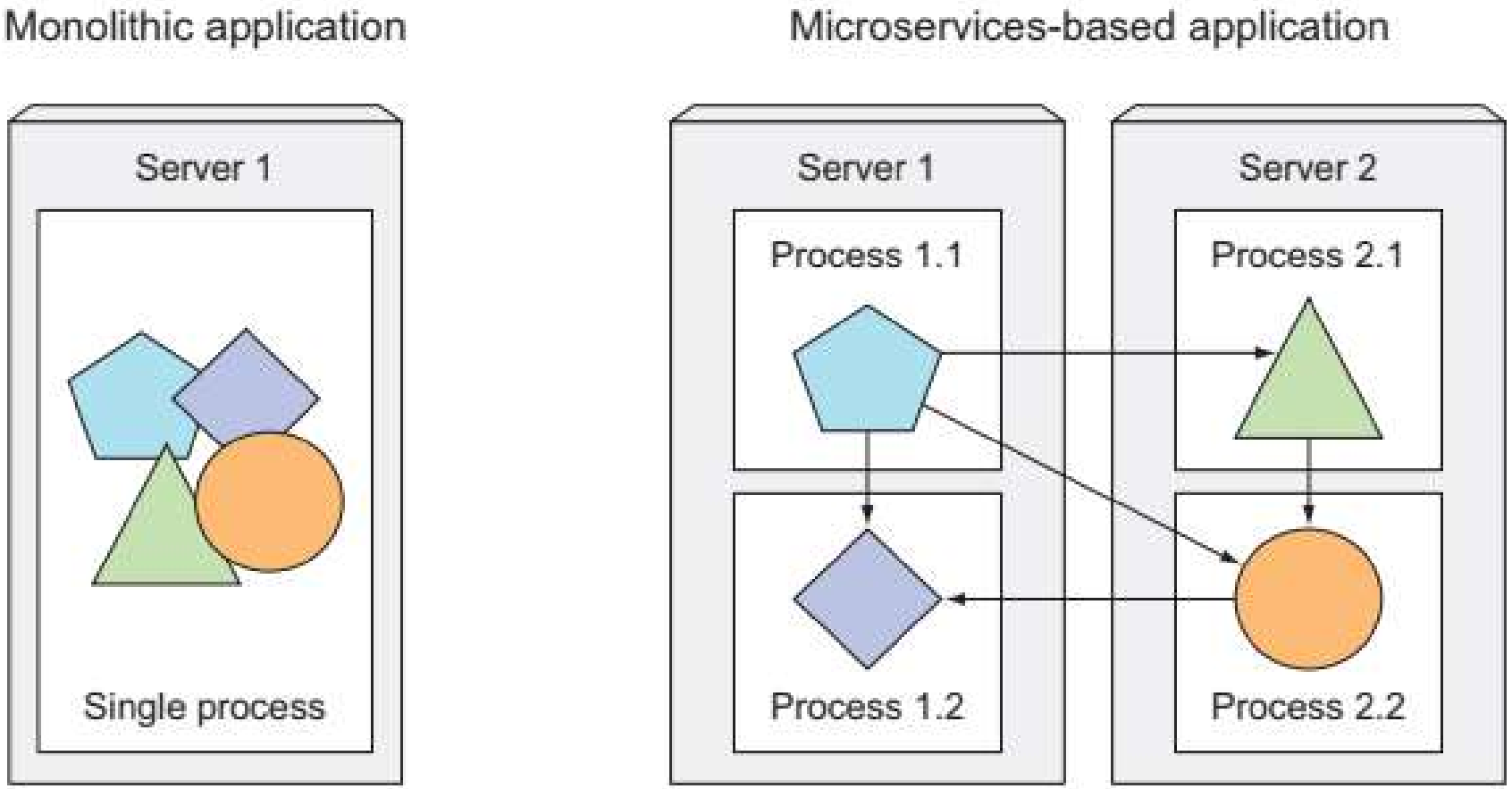
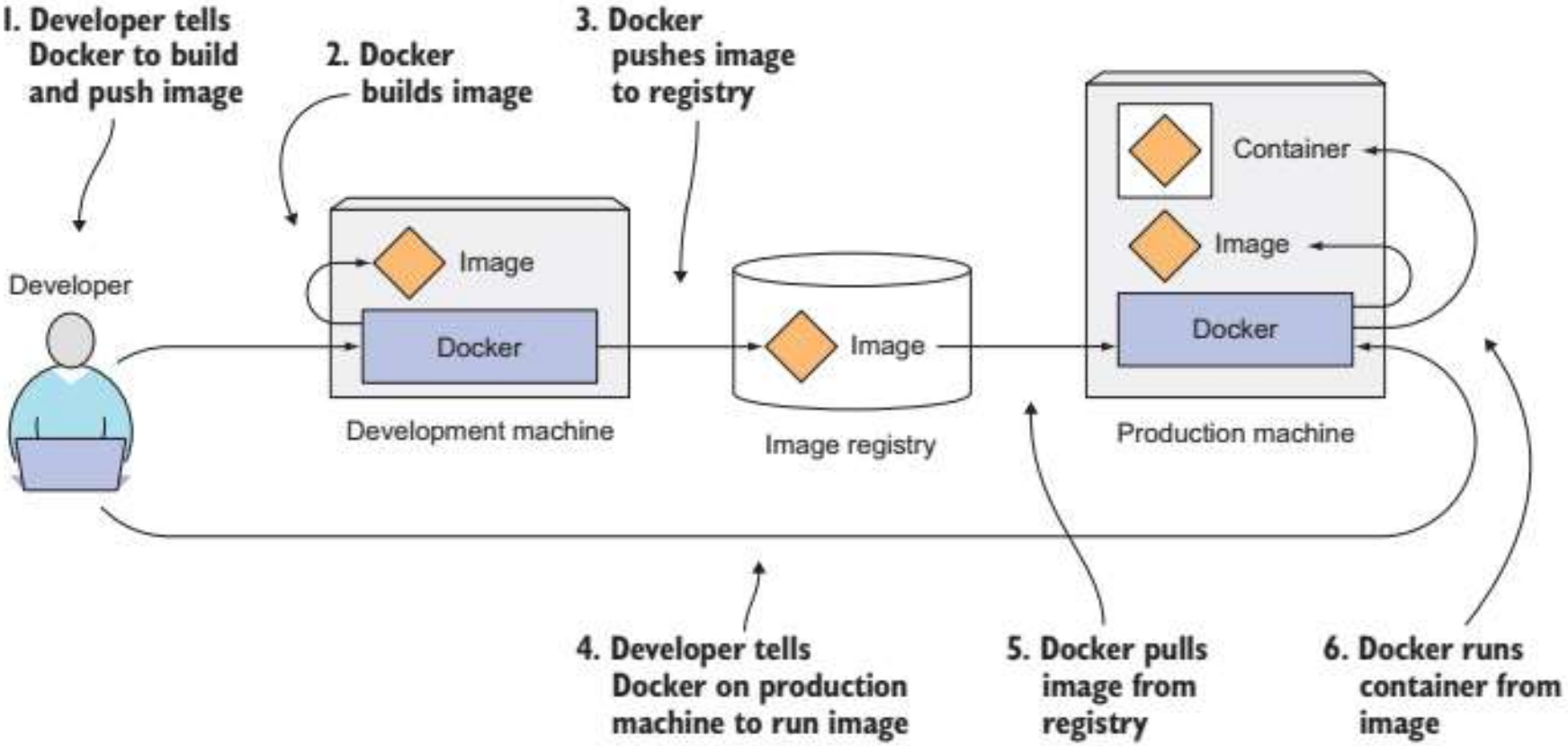


Figure 1.1 Components inside a monolithic application vs. standalone microservices


1. Developer tells Docker to build and push image
2. Docker builds image
3. Docker pushes image to registry
Developer
Image
Docker
Development machine
Image
Image registry
Container
Image
Docker
Production machine
4. Developer tells Docker on production machine to run image
5. Docker pulls image from registry
6. Docker runs container from image

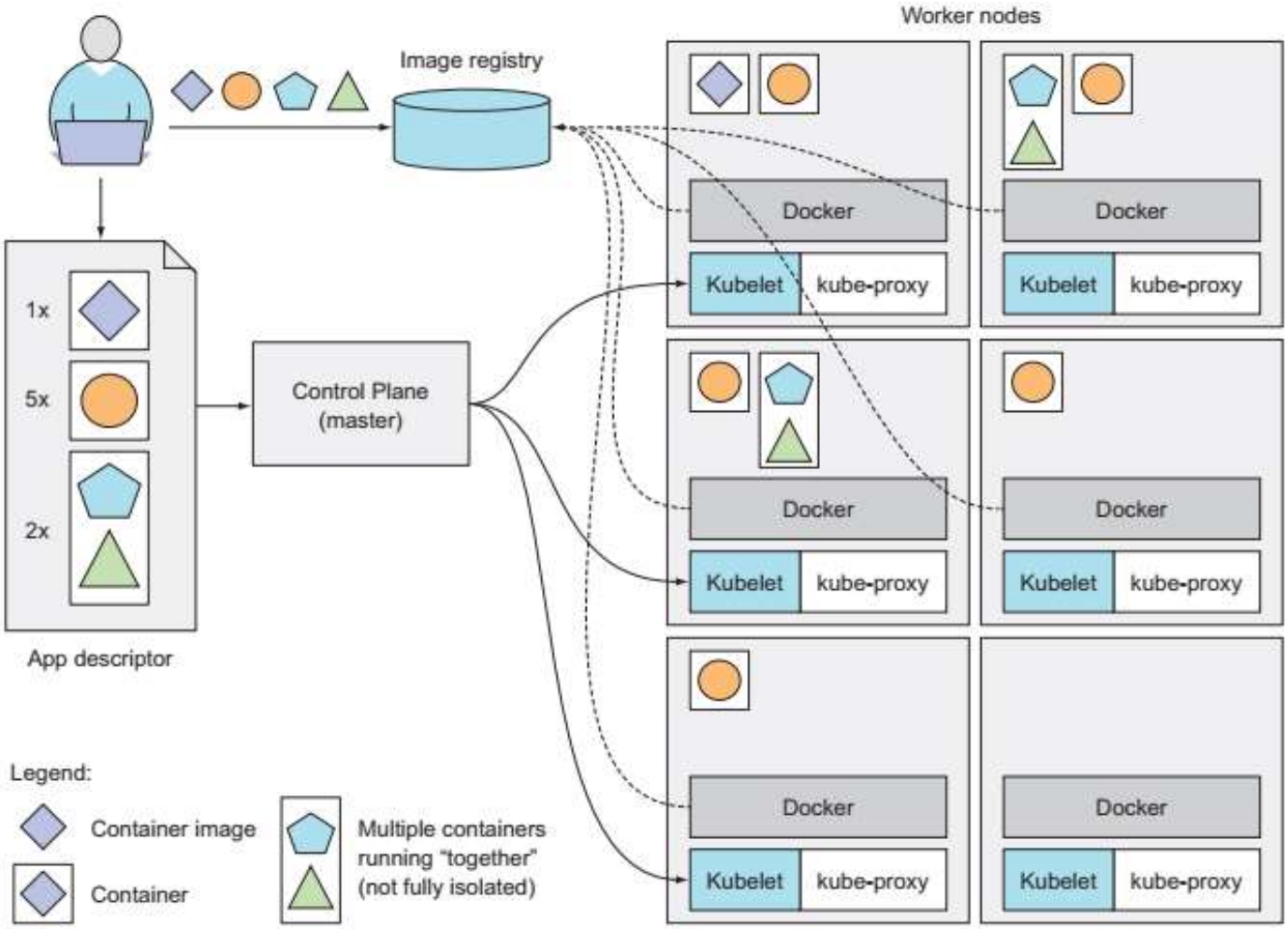


Figure 1.10 A basic overview of the Kubernetes architecture and an application running on top of it

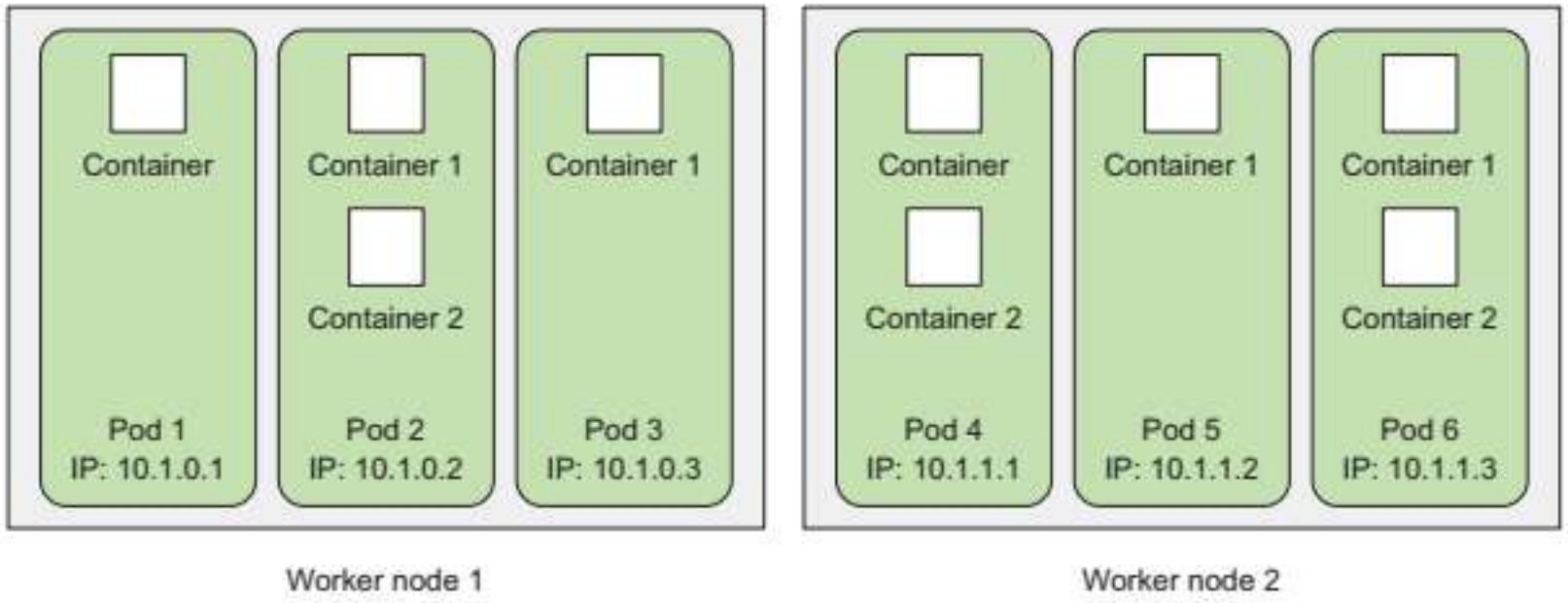


Figure 2.5 The relationship between containers, pods, and physical worker nodes

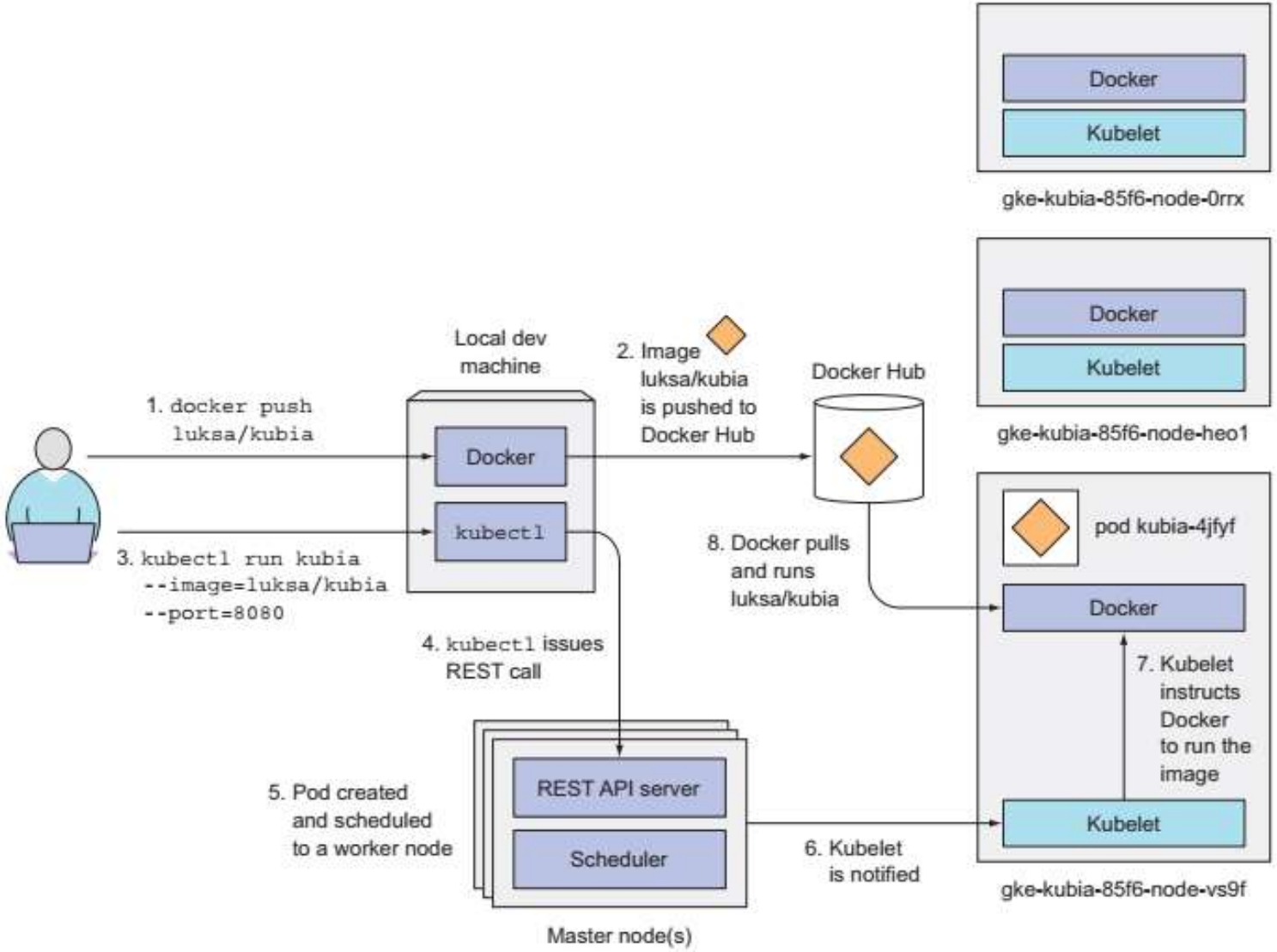


Figure 2.6 Running the `luksa/kubia` container image in Kubernetes

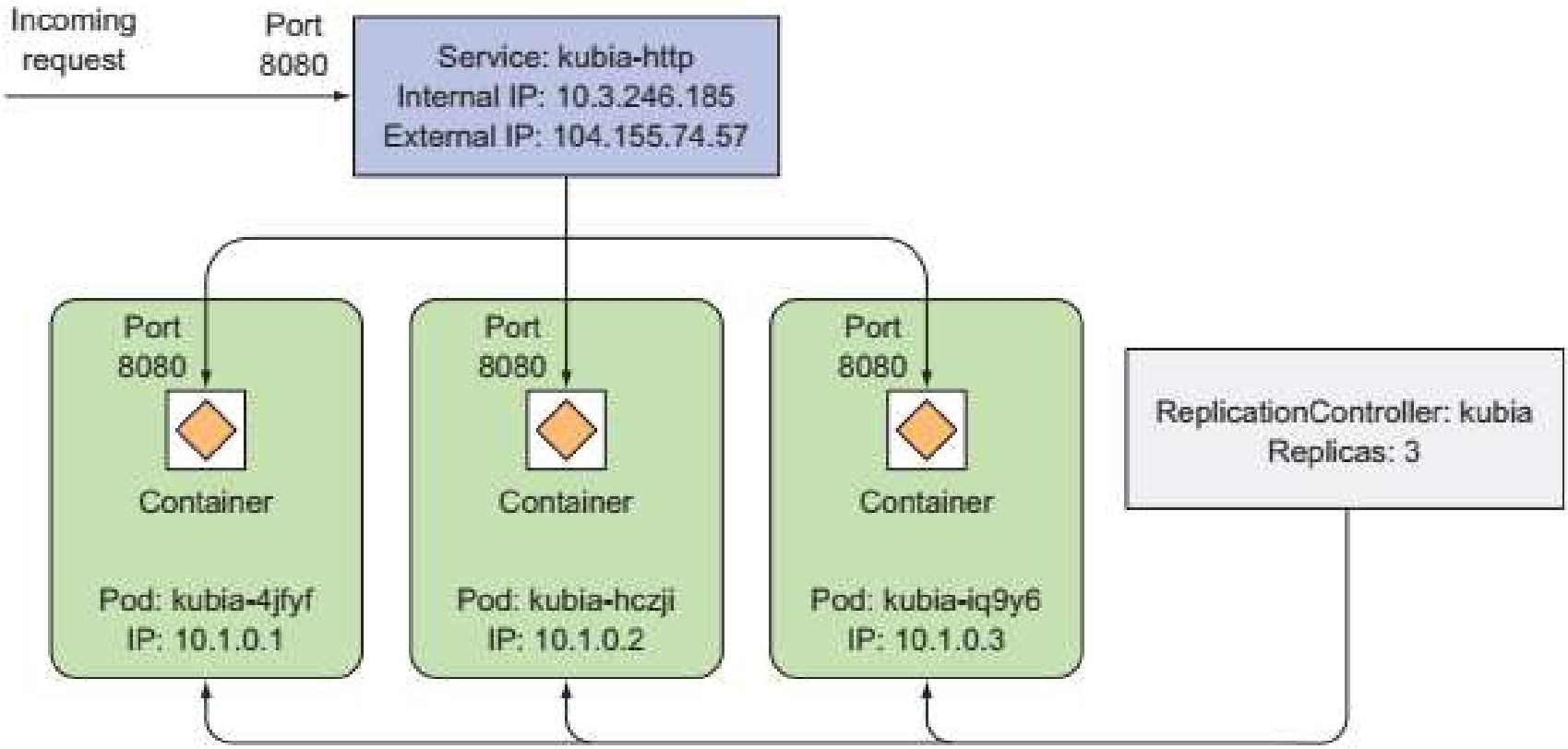
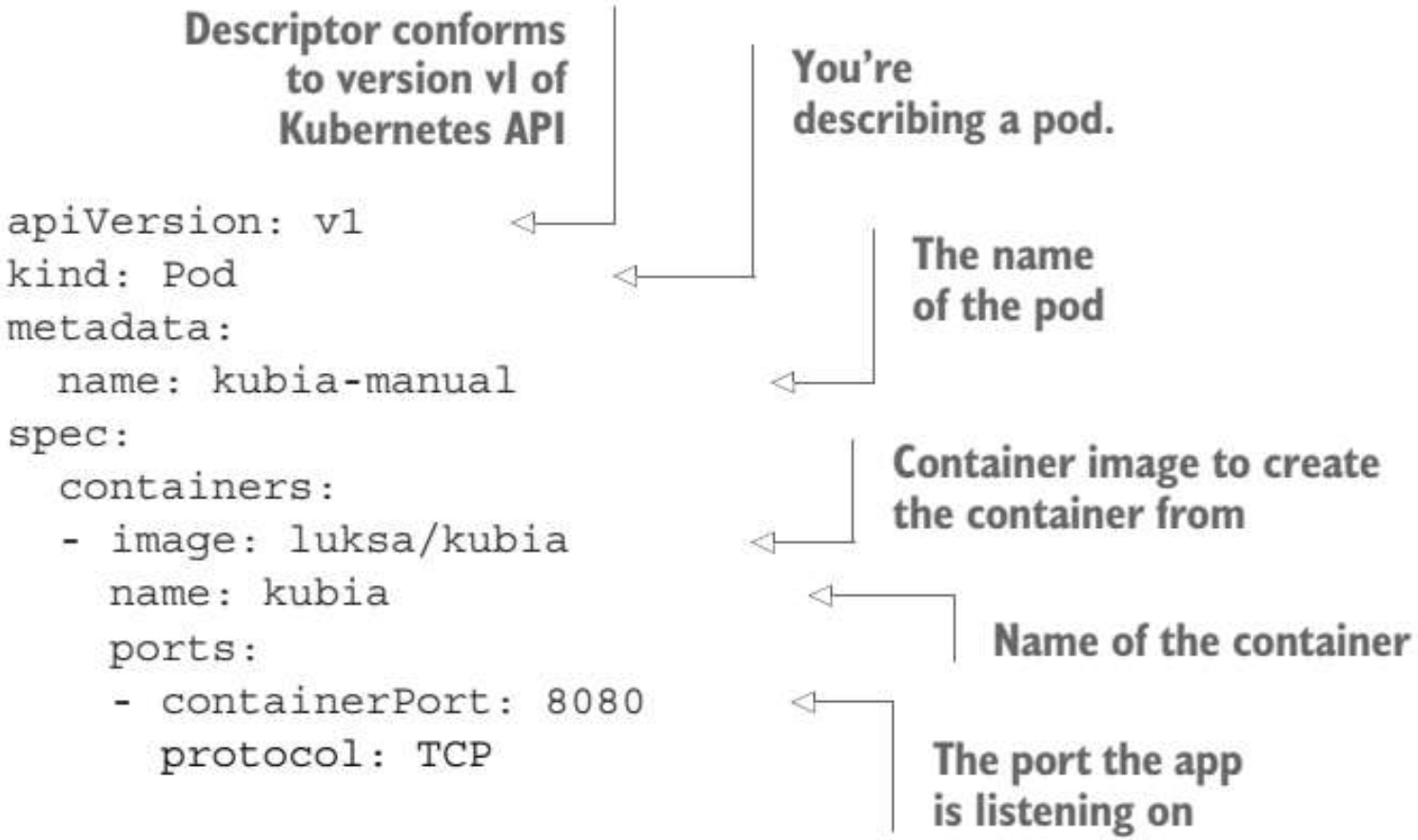


Figure 2.8 Three instances of a pod managed by the same ReplicationController and exposed through a single service IP and port.

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




```
apiVersion: v1
kind: Pod
metadata:
  name: kubia-manual
spec:
  containers:
  - image: luksa/kubia
    name: kubia
    ports:
    - containerPort: 8080
      protocol: TCP
```

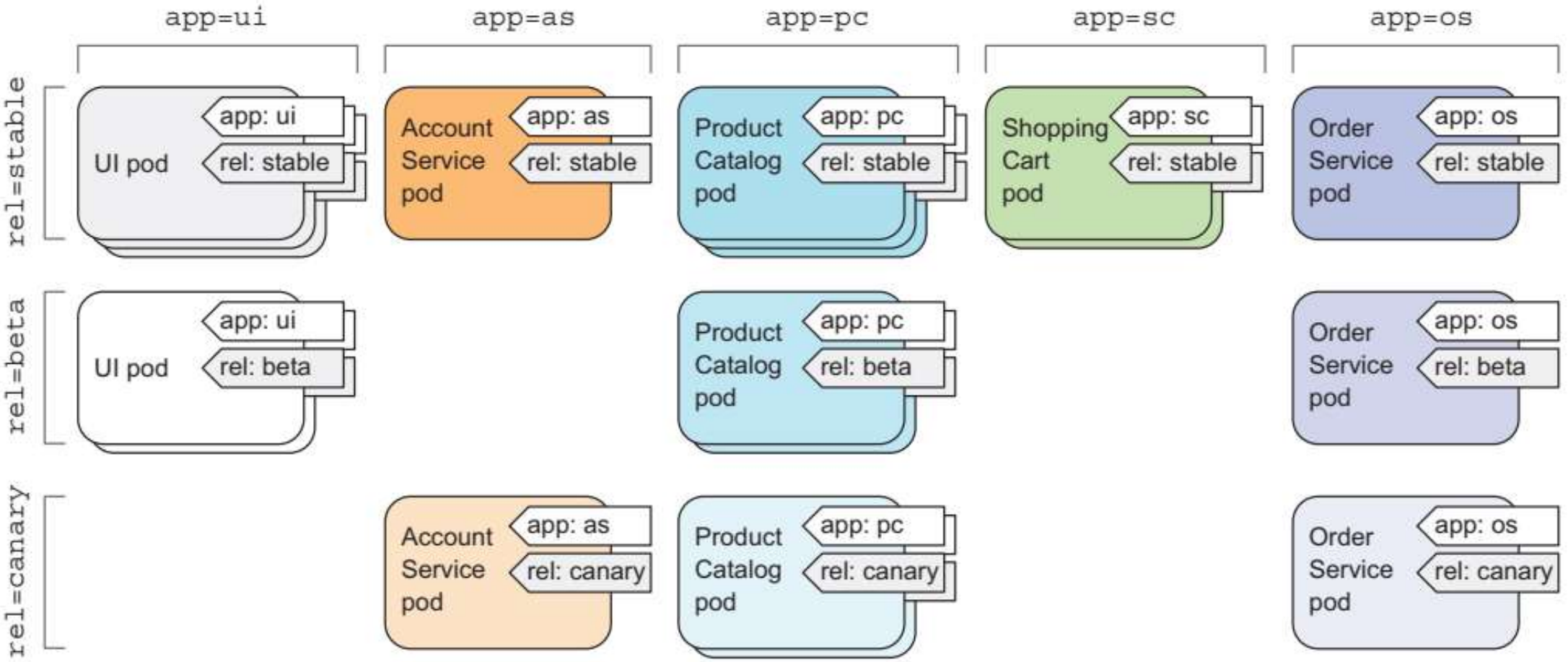


Figure 3.7 Organizing pods in a microservices architecture with pod labels

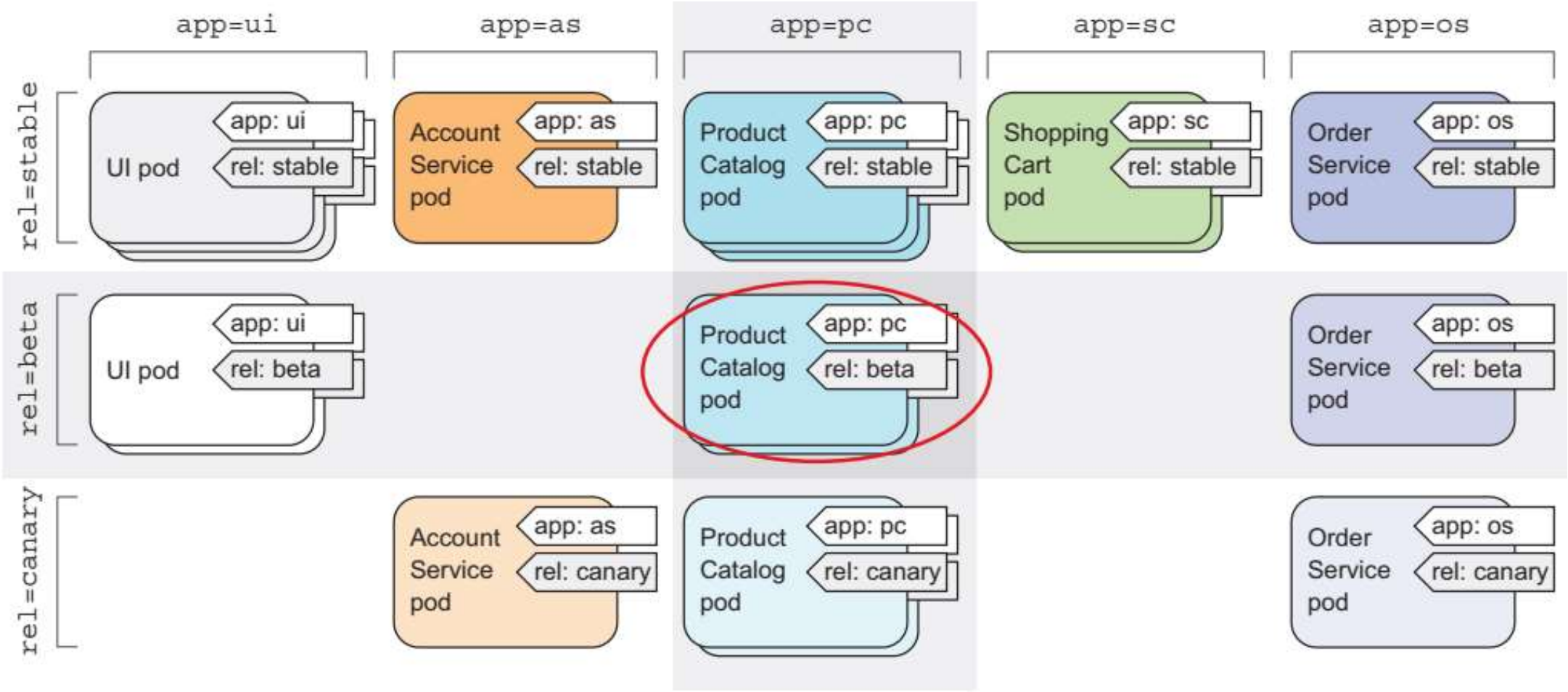


Figure 3.9 Selecting pods with multiple label selectors

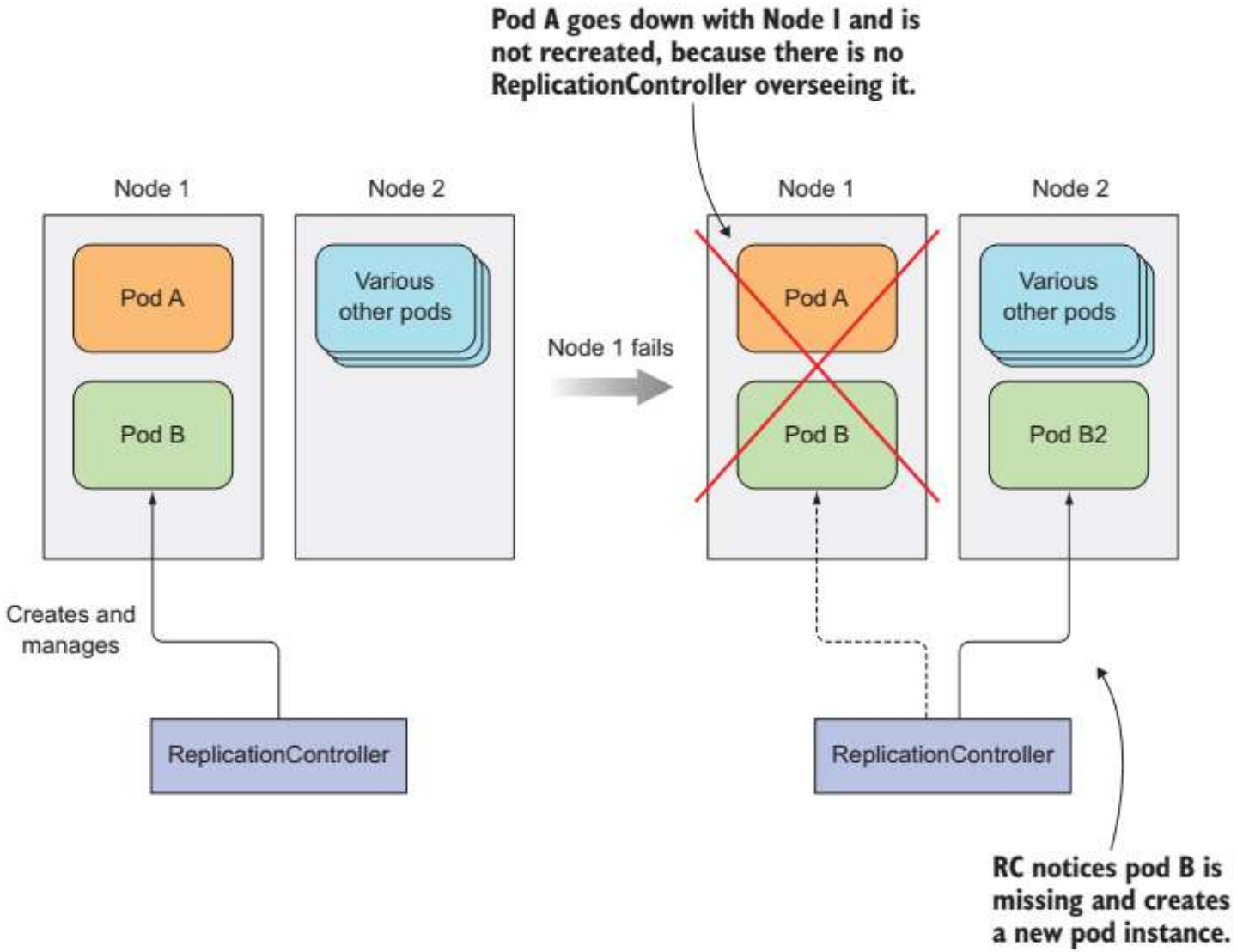


Figure 4.1 When a node fails, only pods backed by a ReplicationController are recreated.

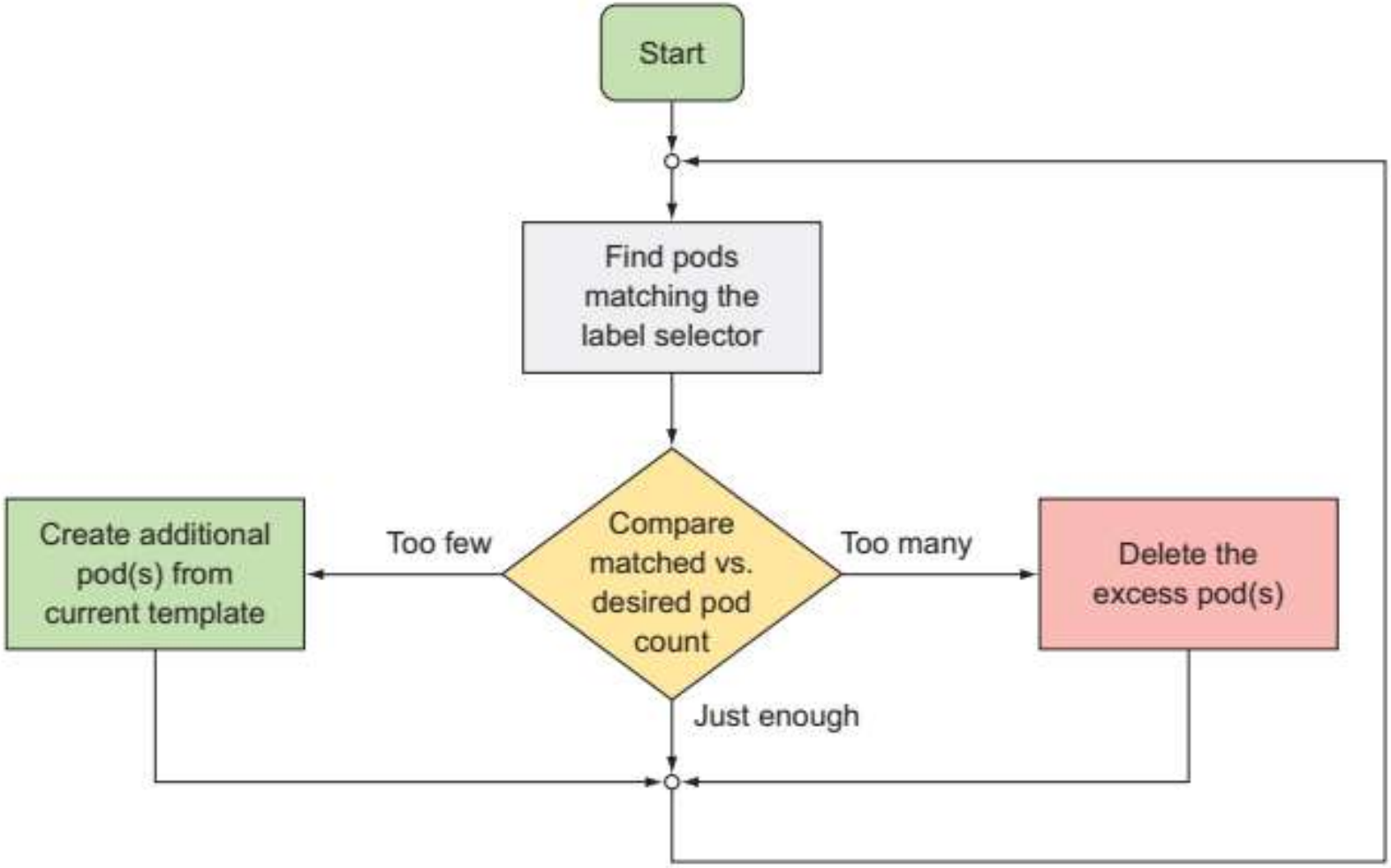


Figure 4.2 A ReplicationController's reconciliation loop

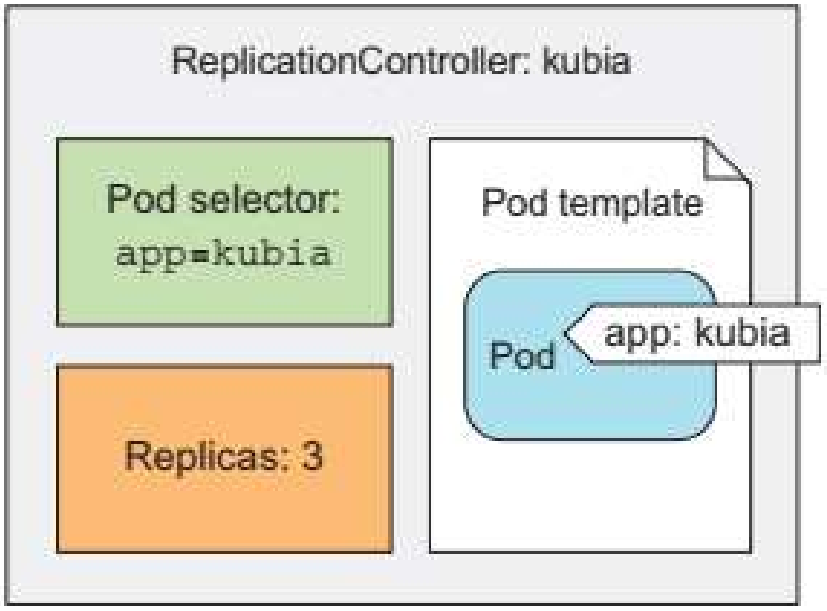


Figure 4.3 The three key parts of a ReplicationController (pod selector, replica count, and pod template)

Services

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.

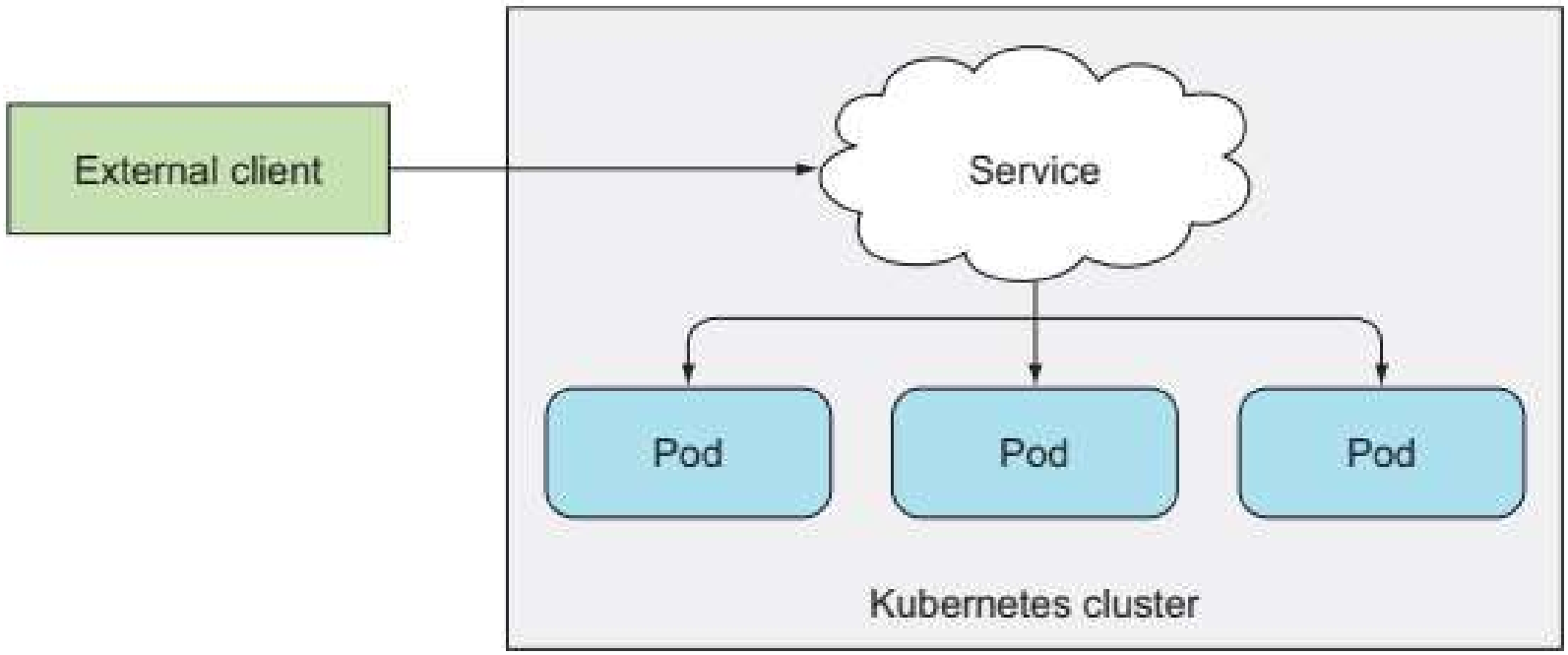


Figure 5.5 Exposing a service to external clients

Thanks