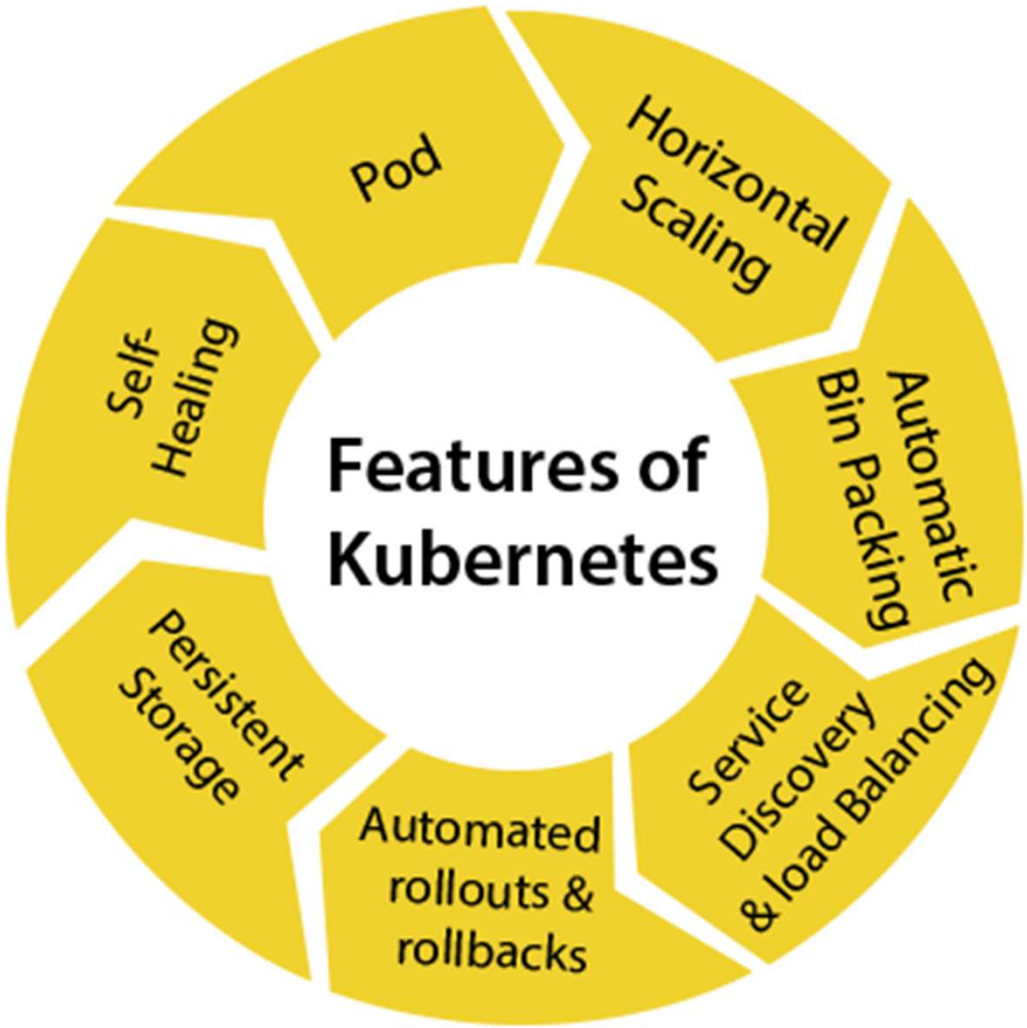
Introduction to Kubernetes

Features


Features of Kubernetes
Pod
Horizontal Scaling
Automatic Bin Packing
Service Discovery & load Balancing
Automated rollouts & rollbacks
Persistent Storage
Self-Healing

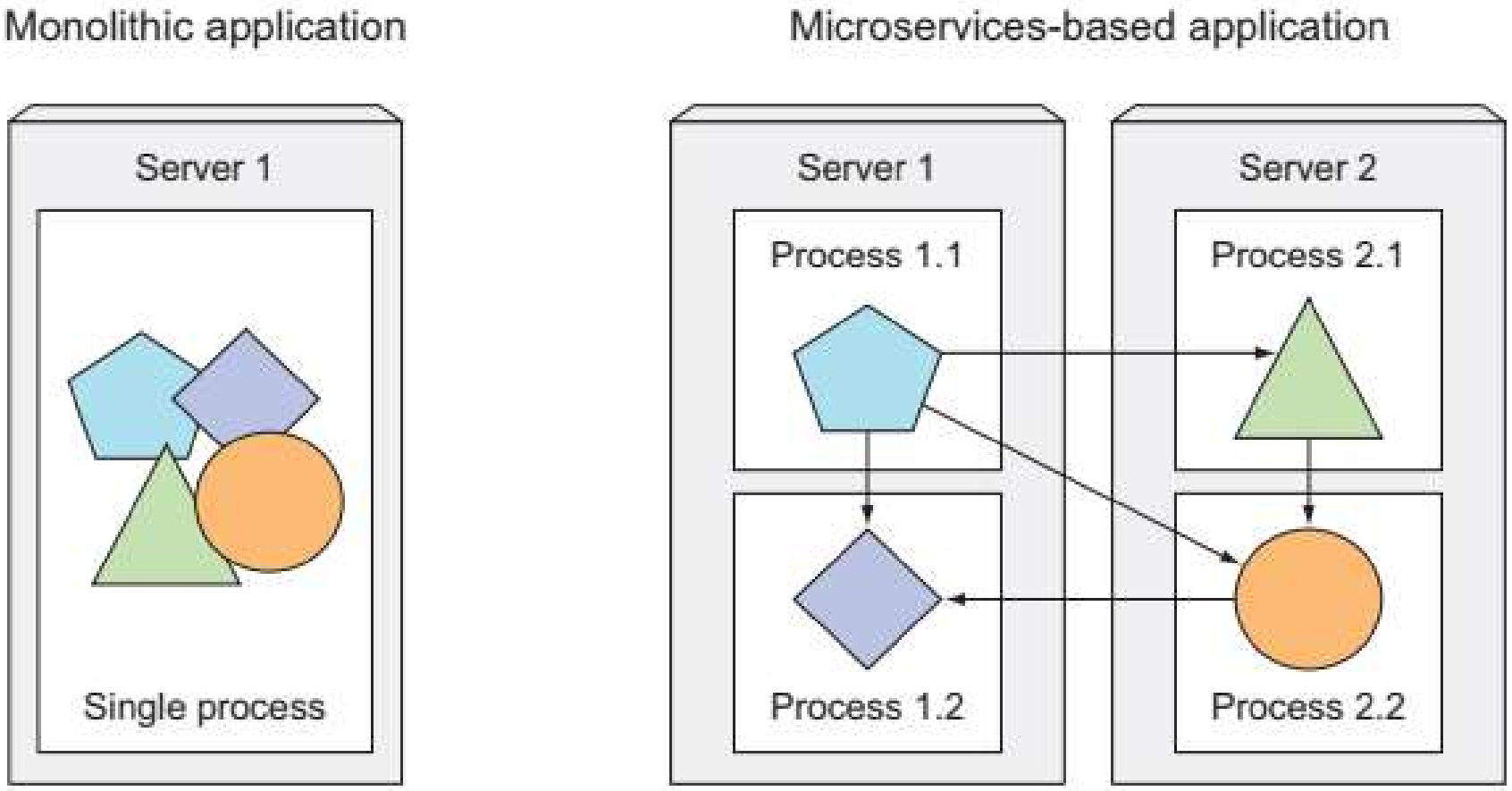
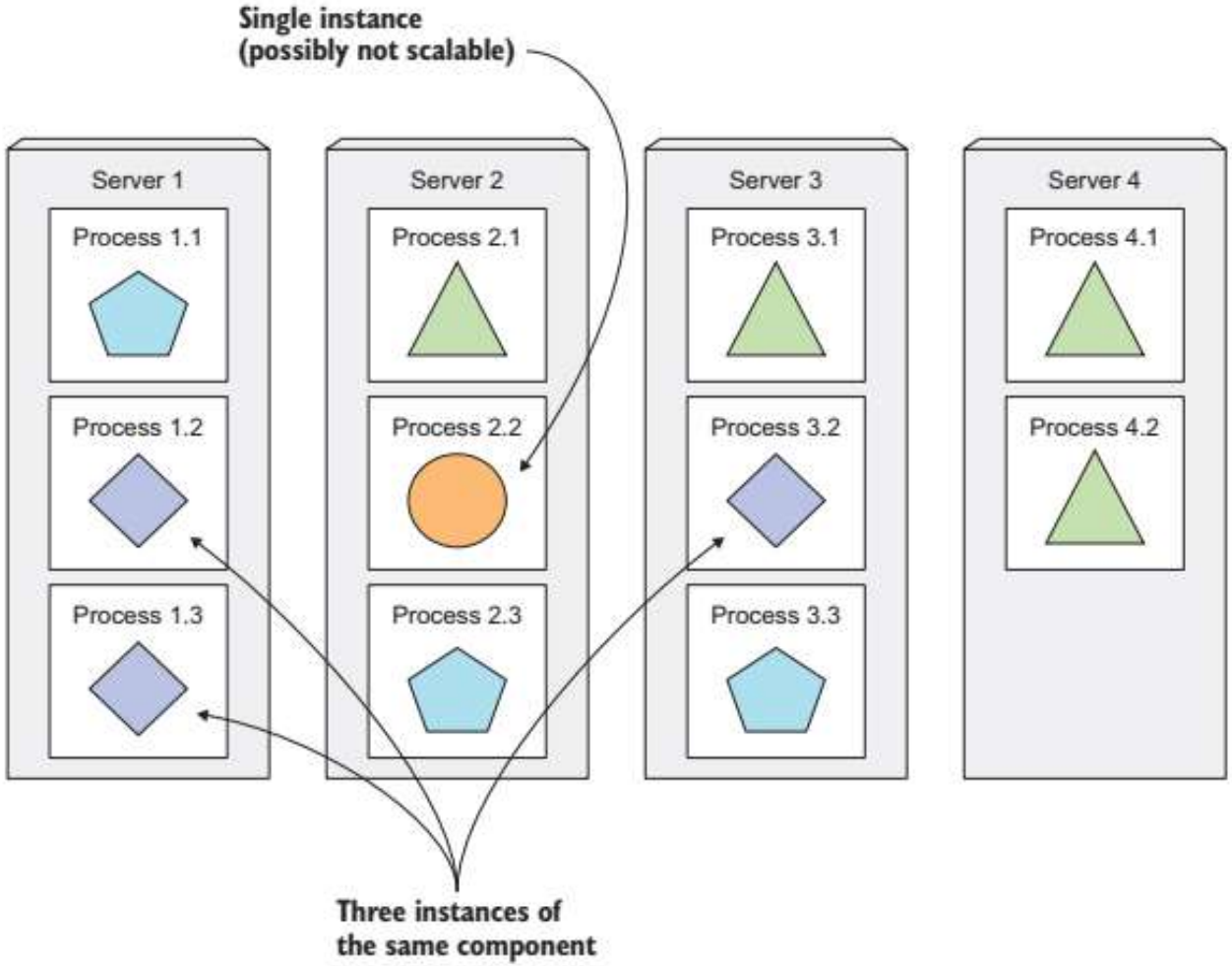
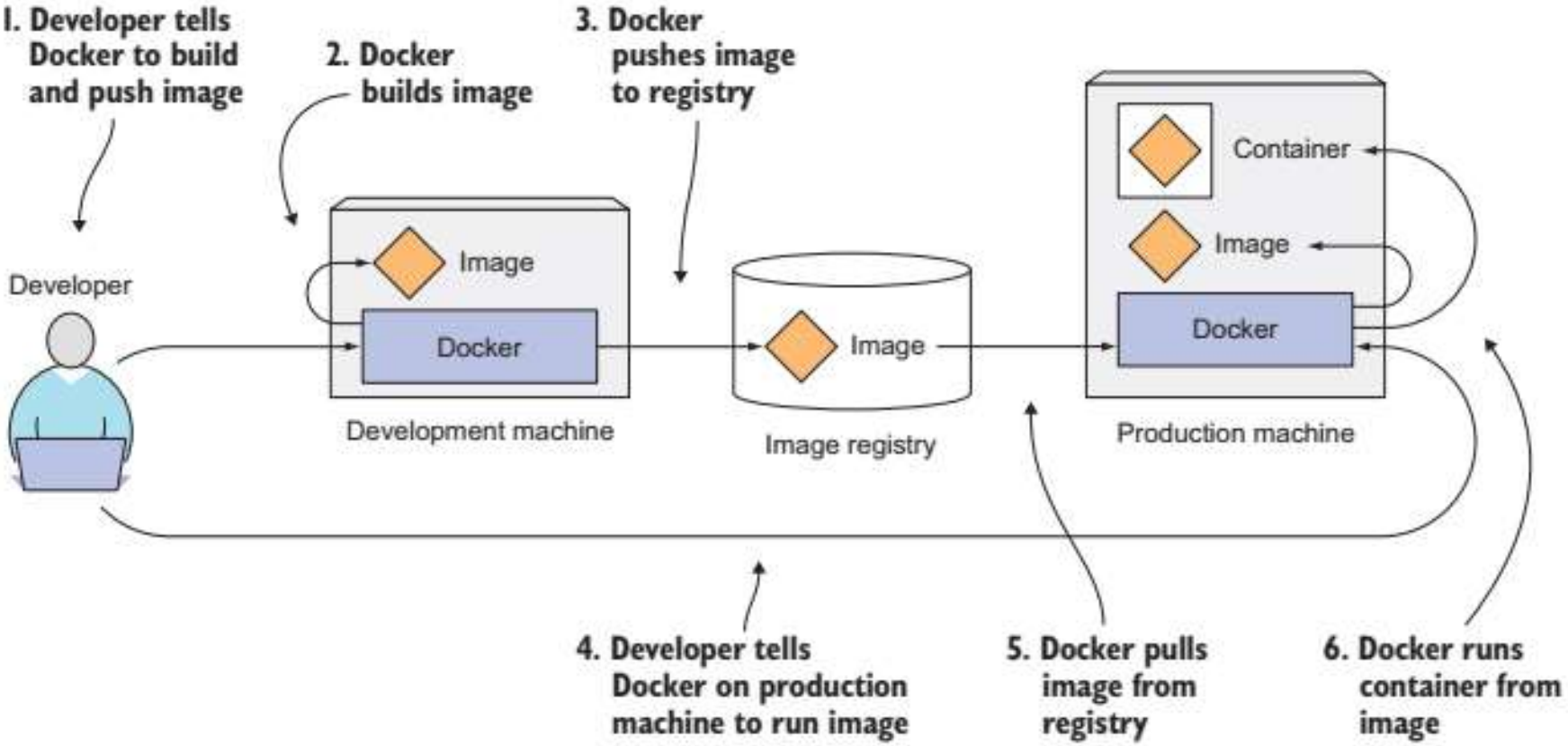
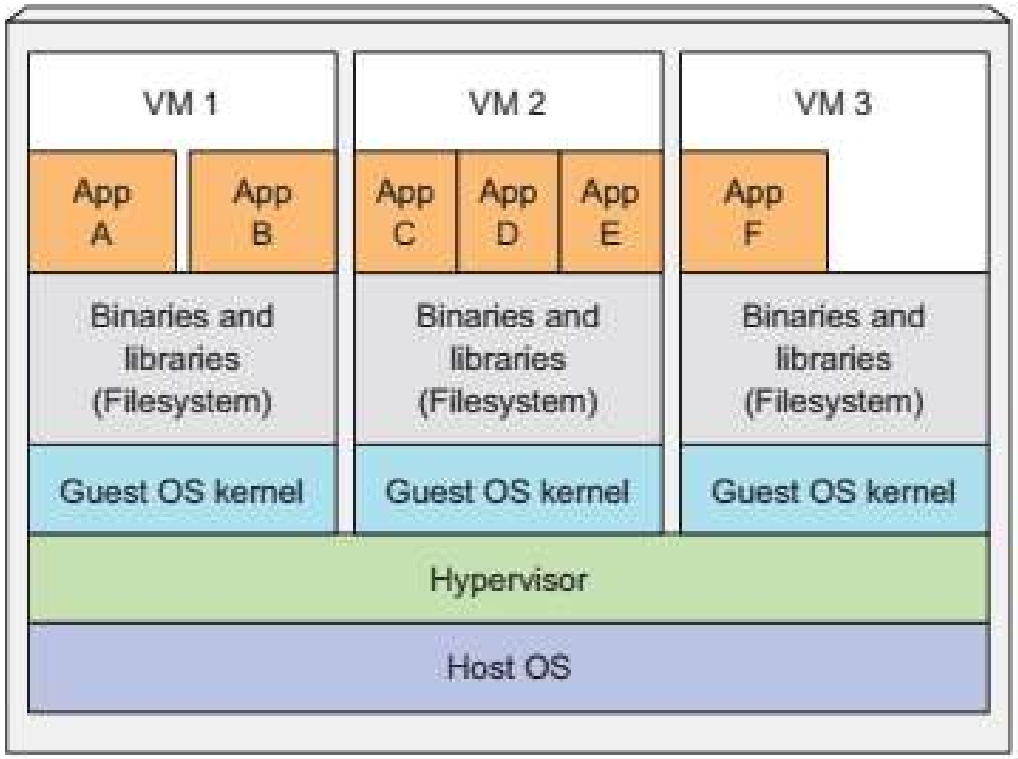


Figure 1.1 Components inside a monolithic application vs. standalone microservices

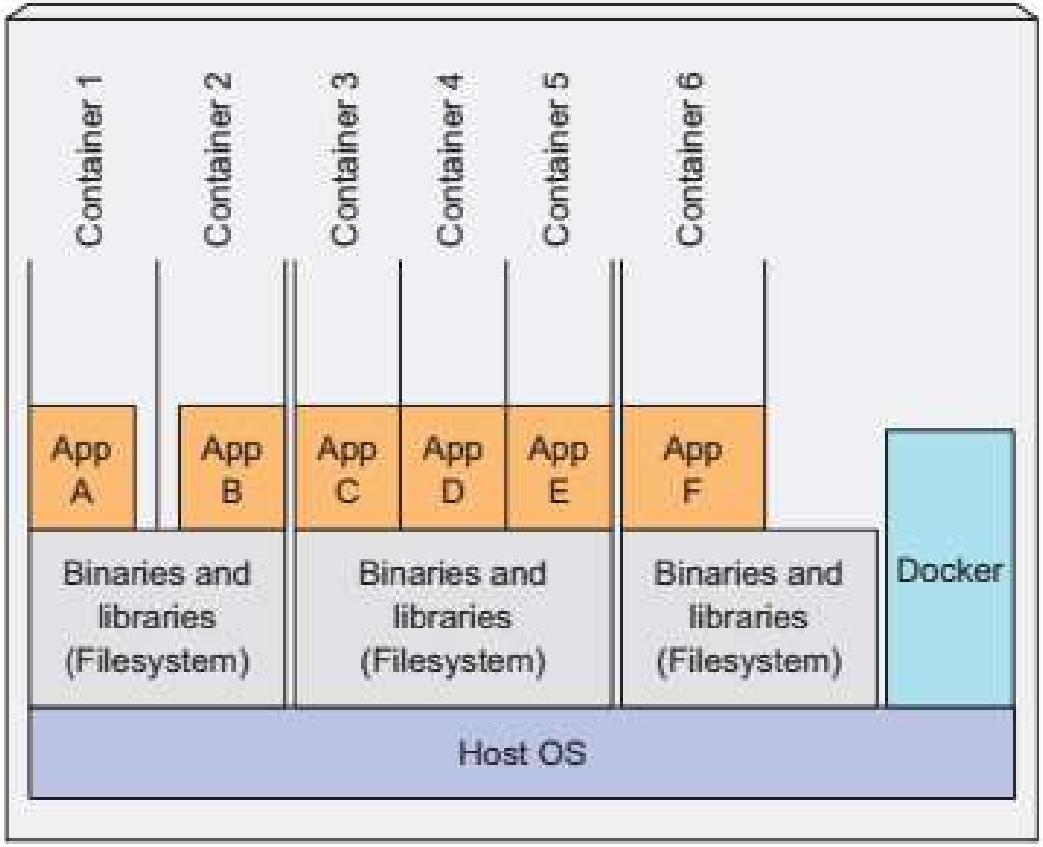

Single instance
(possibly not scalable)
Server 1
Process 1.1
Process 1.2
Process 1.3
Server 2
Process 2.1
Process 2.2
Process 2.3
Server 3
Process 3.1
Process 3.2
Process 3.3
Server 4
Process 4.1
Process 4.2
Three instances of
the same component


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

Host running multiple VMs



Host running multiple Docker containers



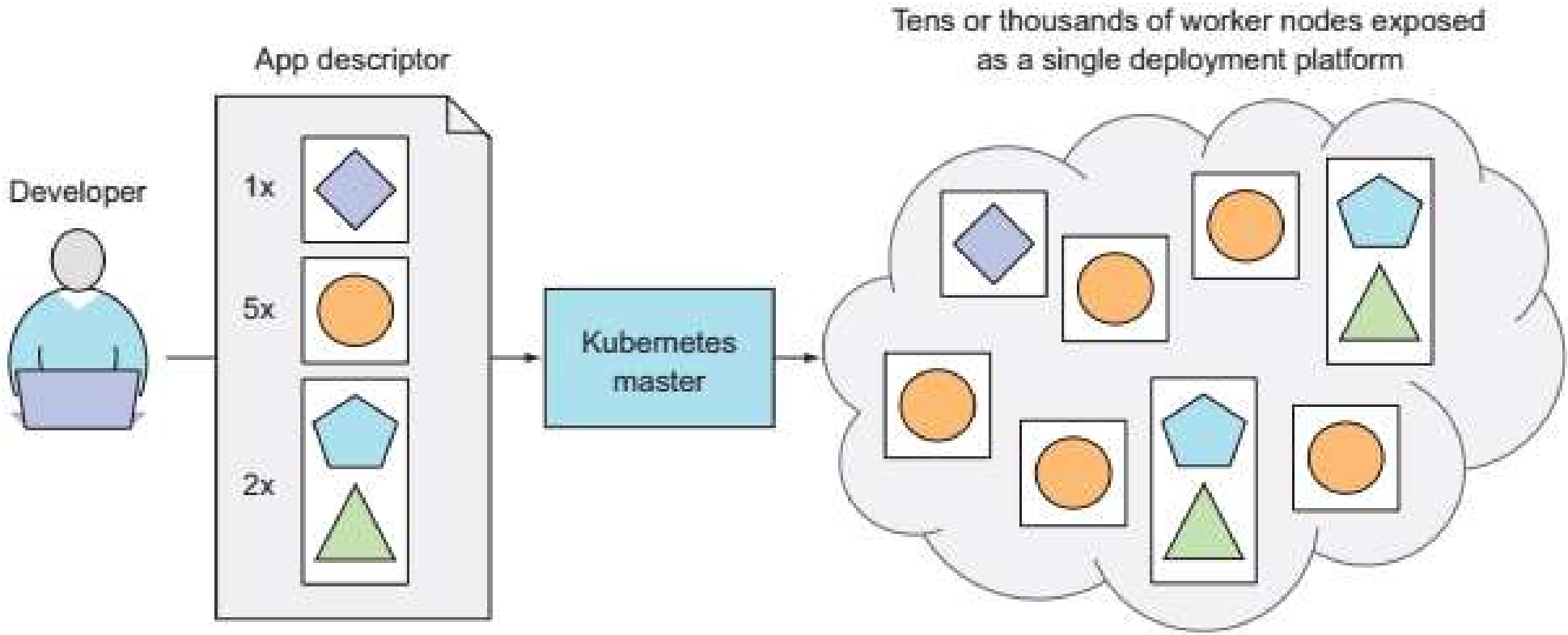


Figure 1.8 Kubernetes exposes the whole datacenter as a single deployment platform.

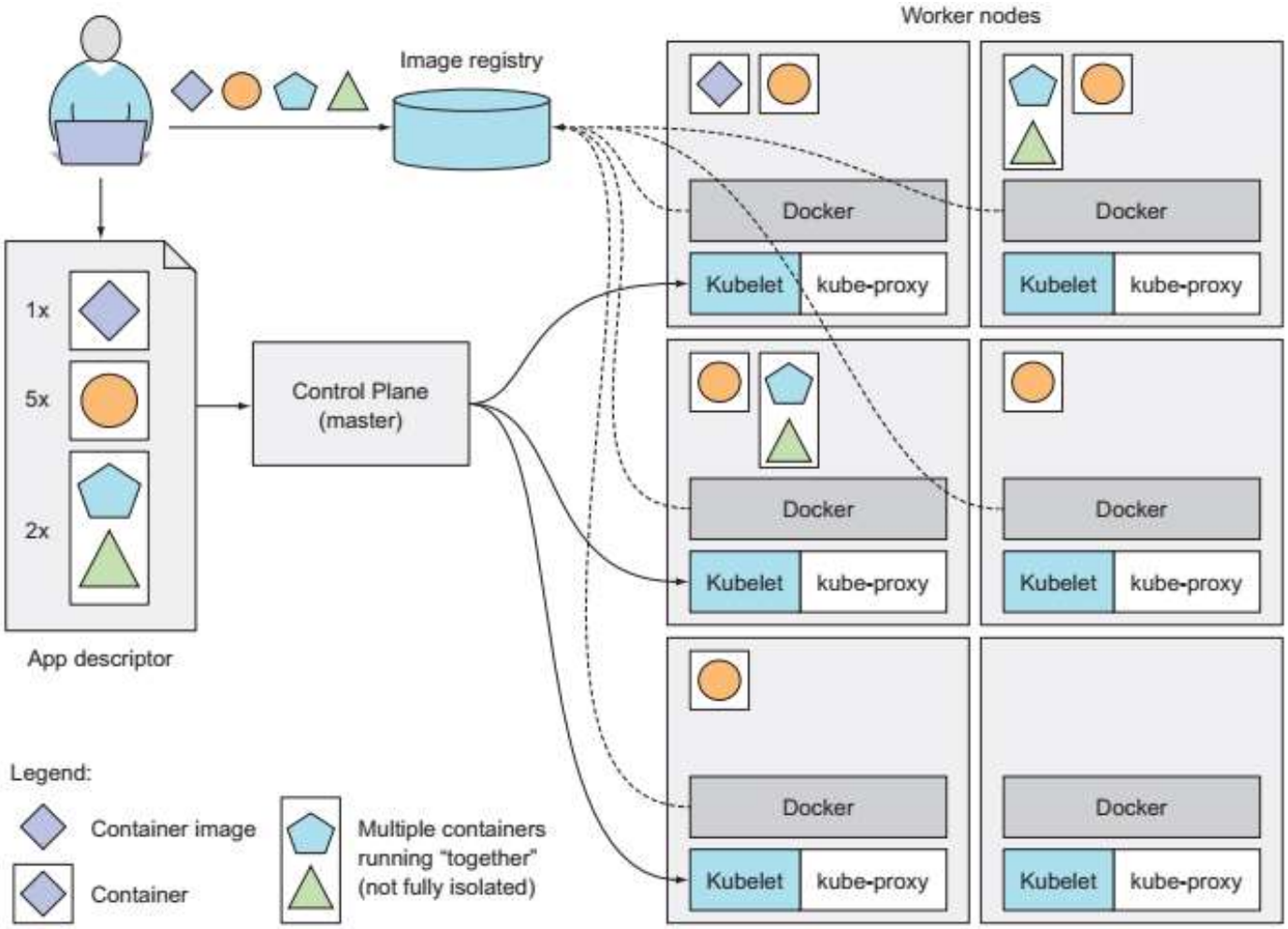


Figure 1.10 A basic overview of the Kubernetes architecture and an application running on top of it

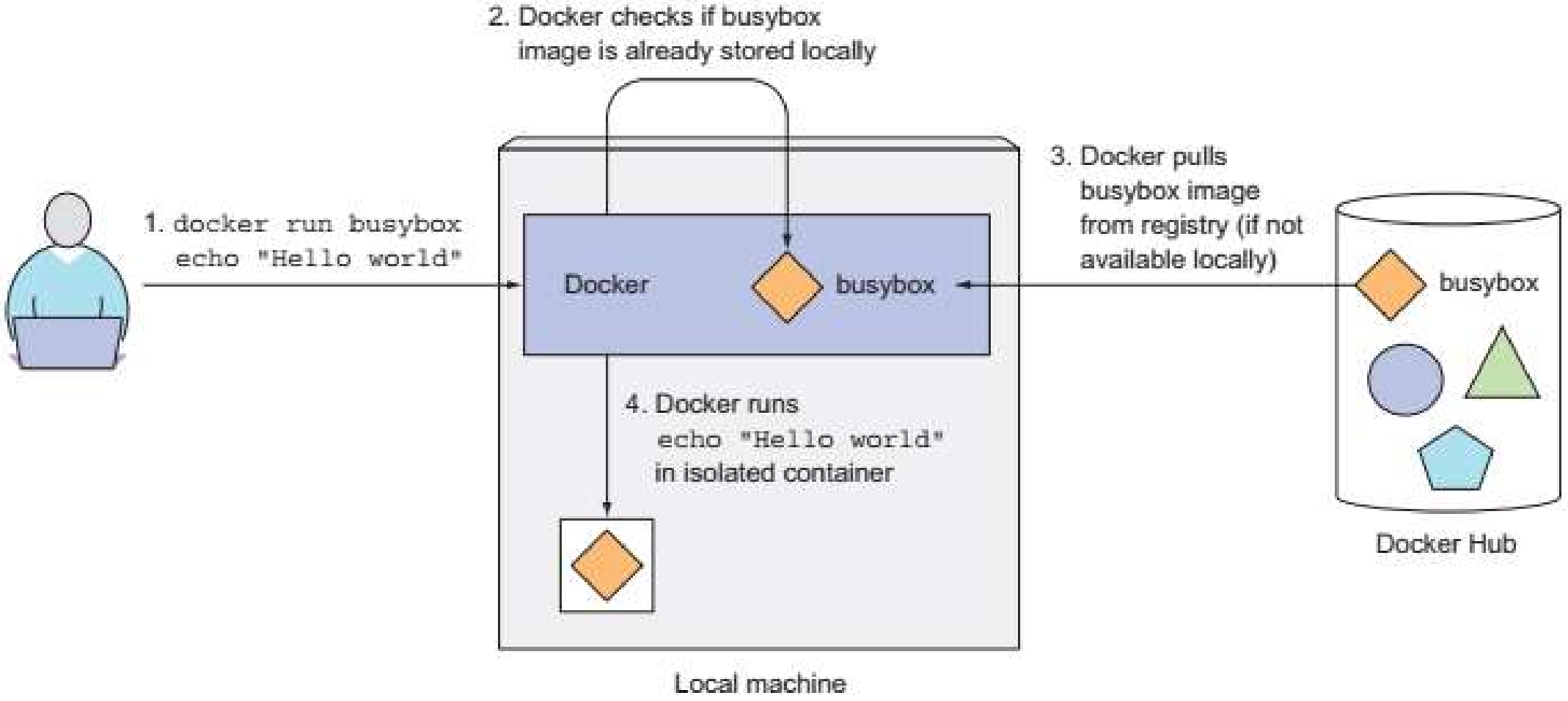


Figure 2.1 Running echo "Hello world" in a container based on the busybox container image

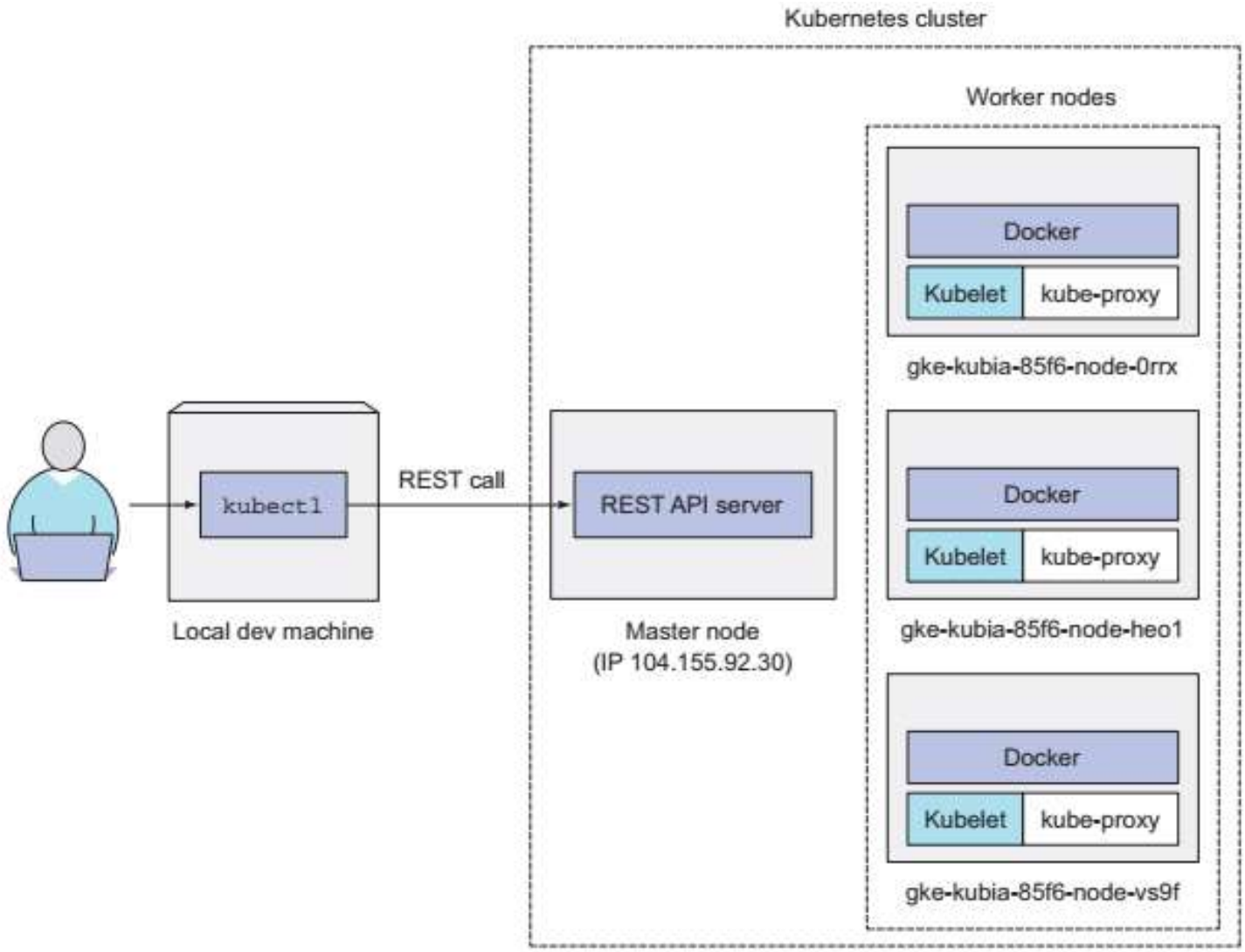


Figure 2.4 How you're interacting with your three-node Kubernetes cluster

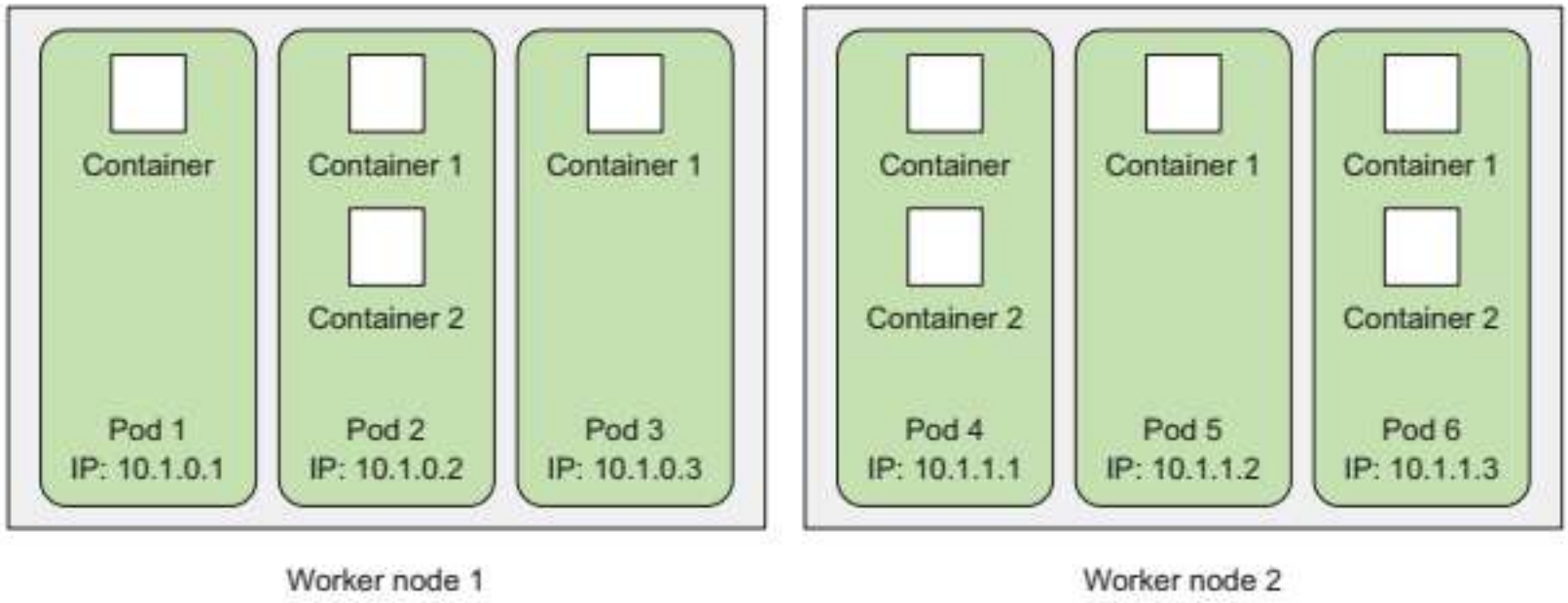
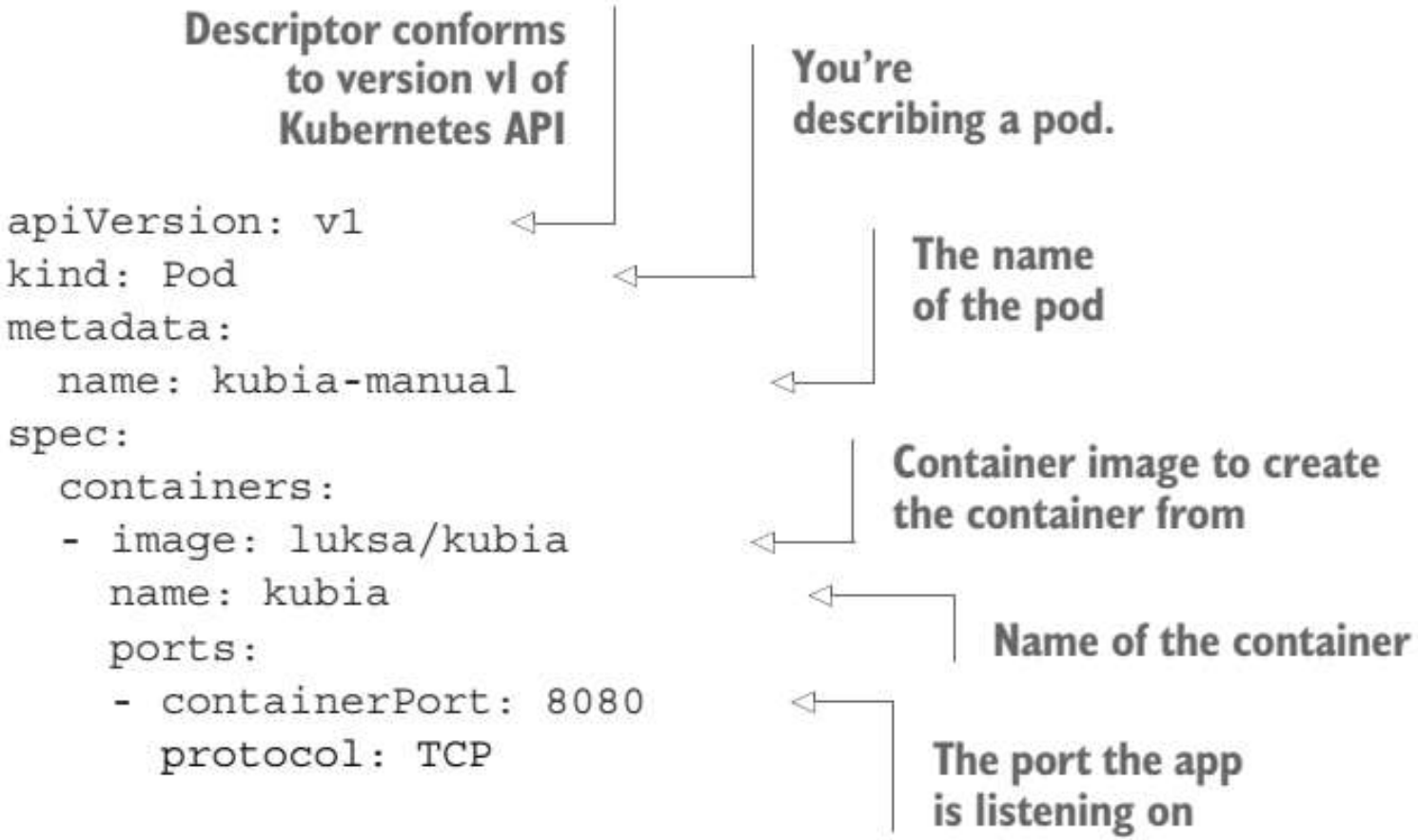


Figure 2.5 The relationship between containers, pods, and physical worker nodes

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
```

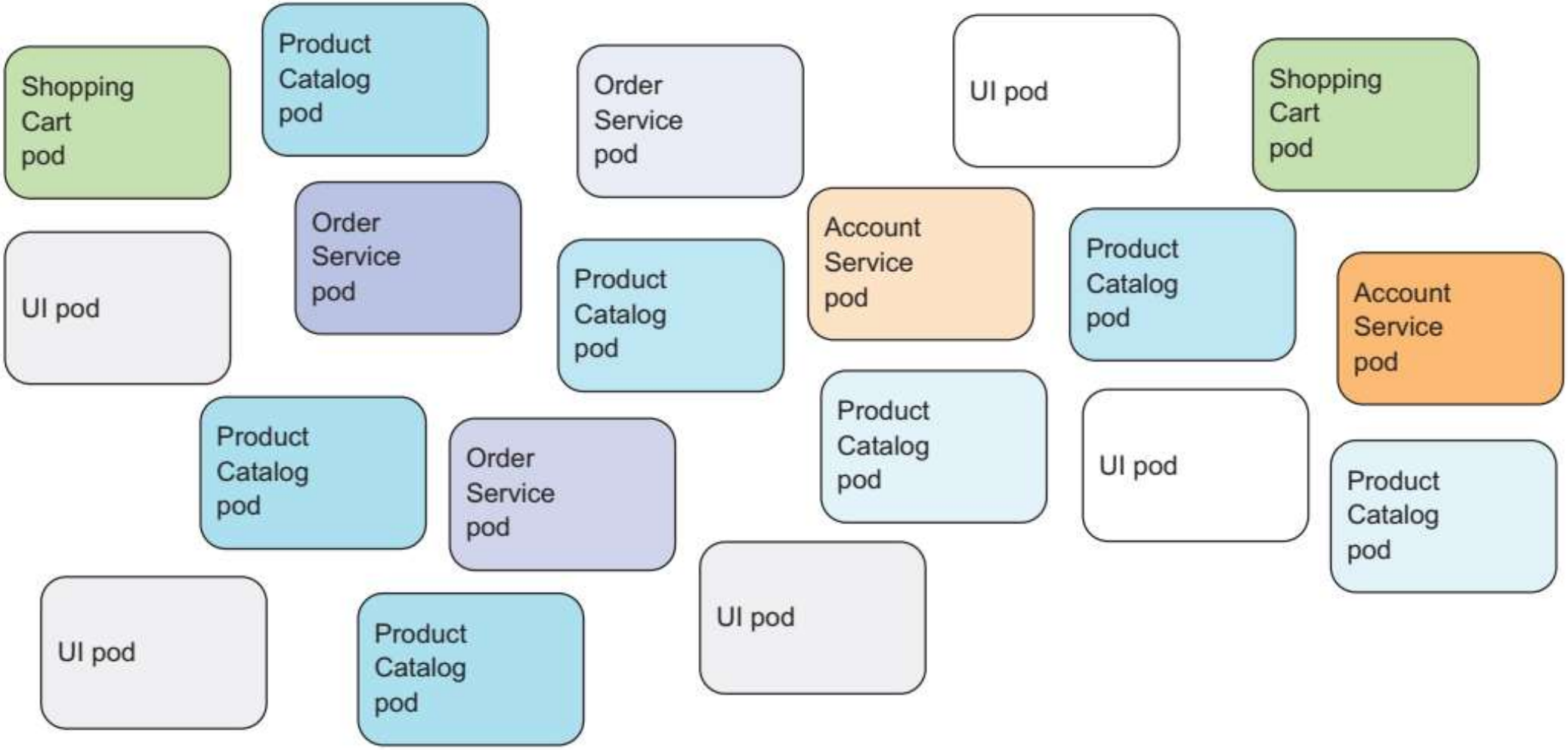


Figure 3.6 Uncategorized pods in a microservices architecture

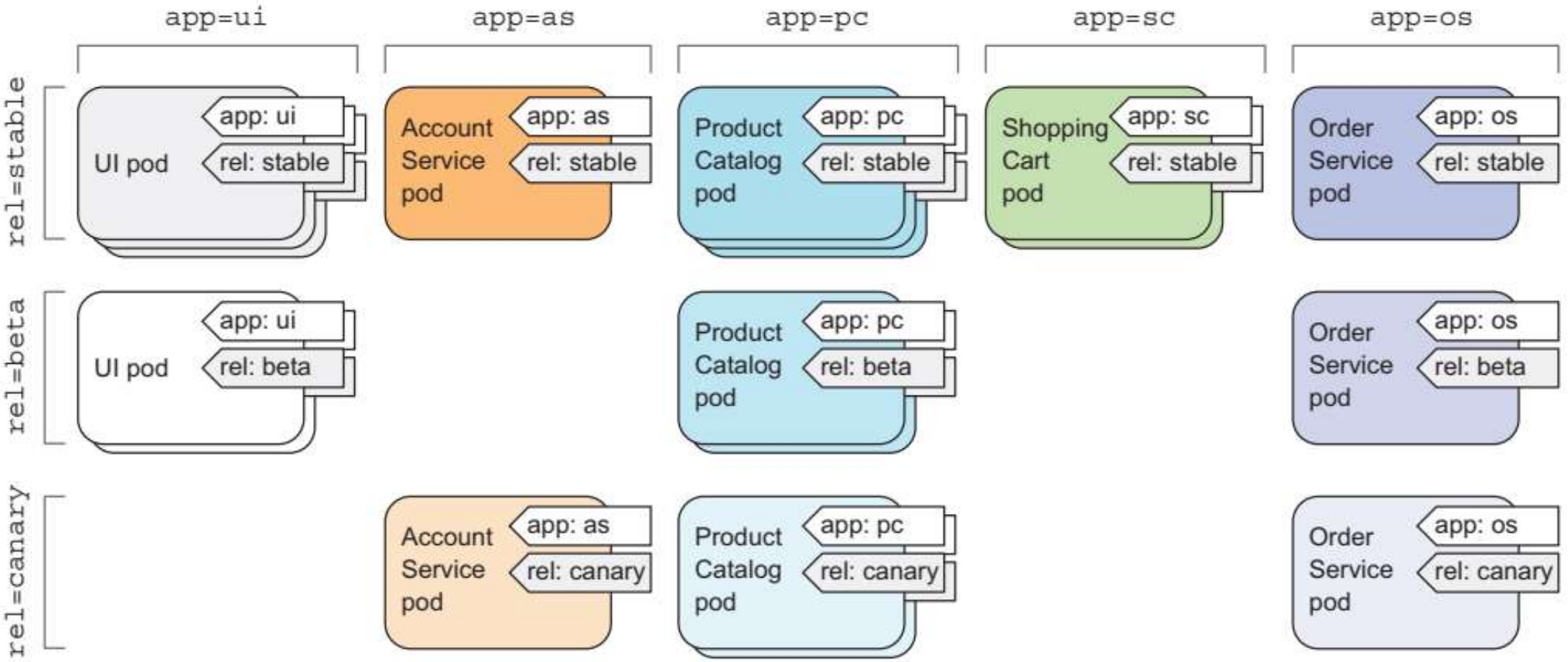


Figure 3.7 Organizing pods in a microservices architecture with pod labels

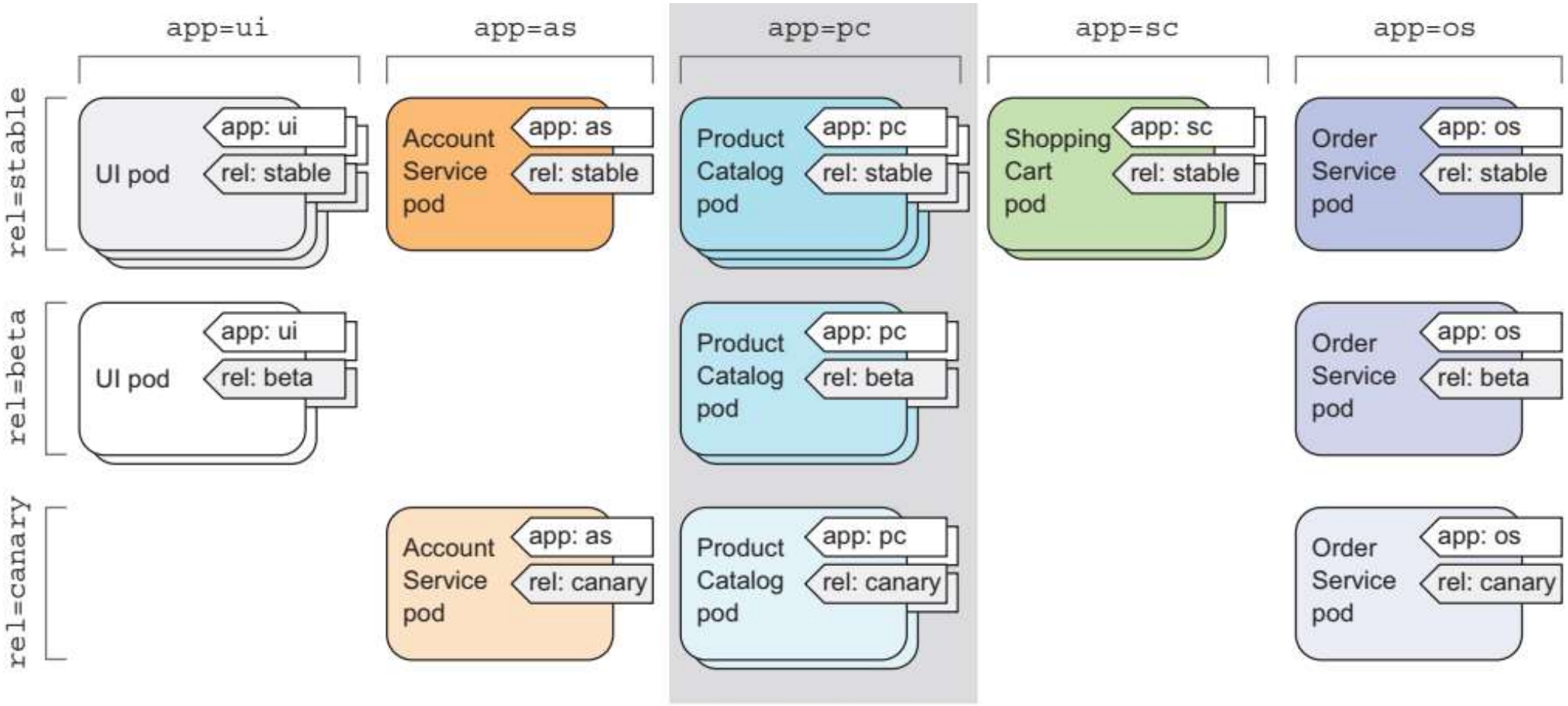


Figure 3.8 Selecting the product catalog microservice pods using the "app=pc" label selector

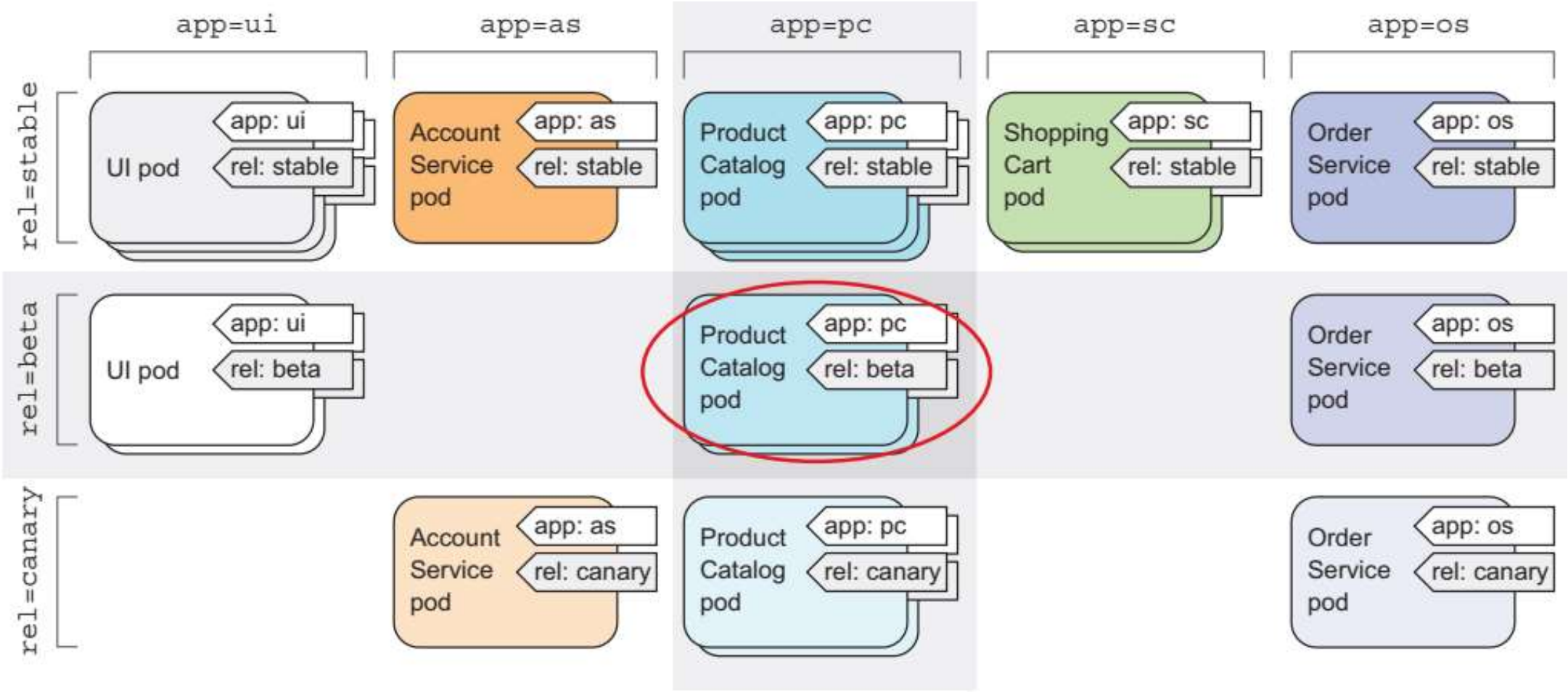
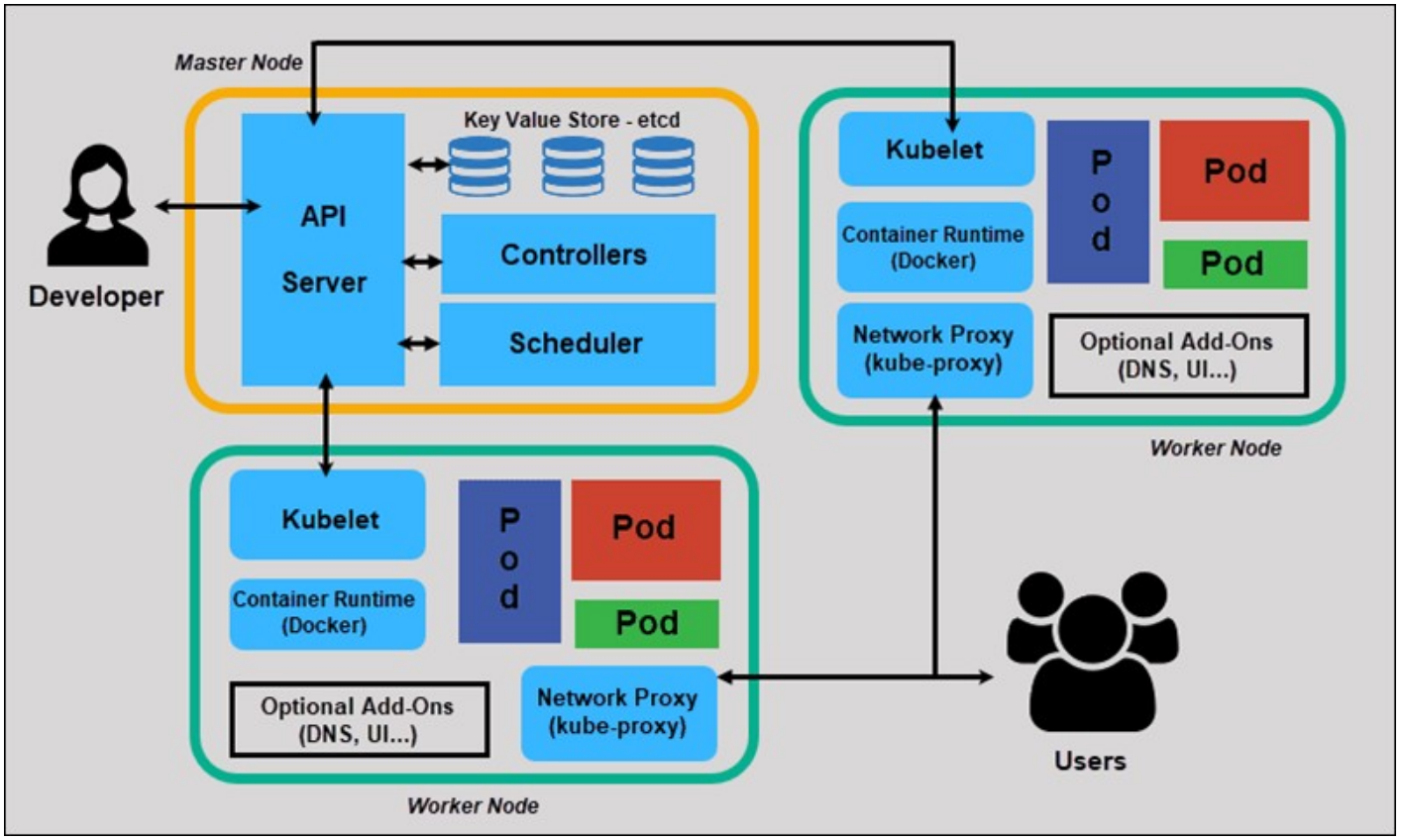


Figure 3.9 Selecting pods with multiple label selectors

Kubernetes Architecture

- A Kubernetes cluster consists of two main components:
  - Master (Control Plane)
  - Worker Nodes.
- Master has following components. These components are responsible for maintaining the state of the cluster:
  - etcd distributed key value store.
  - API Server.
  - Controller Manager
  - Scheduler
- Every worker node consists of the following components.
- These components are responsible for deploying and running the application containers.
  - Kubelet
  - Container Runtime (Docker)

Kubernetes Architecture



Thanks