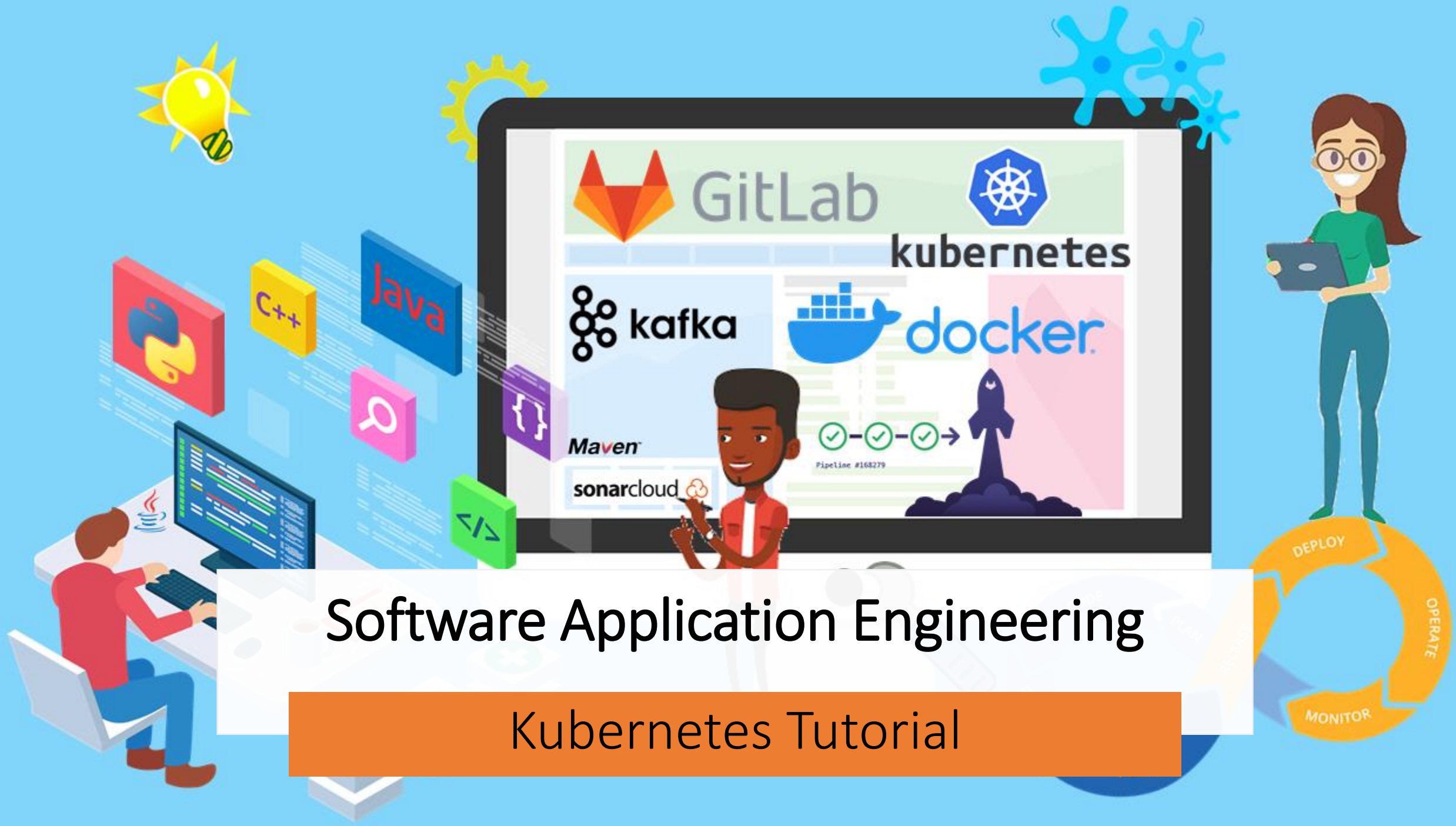
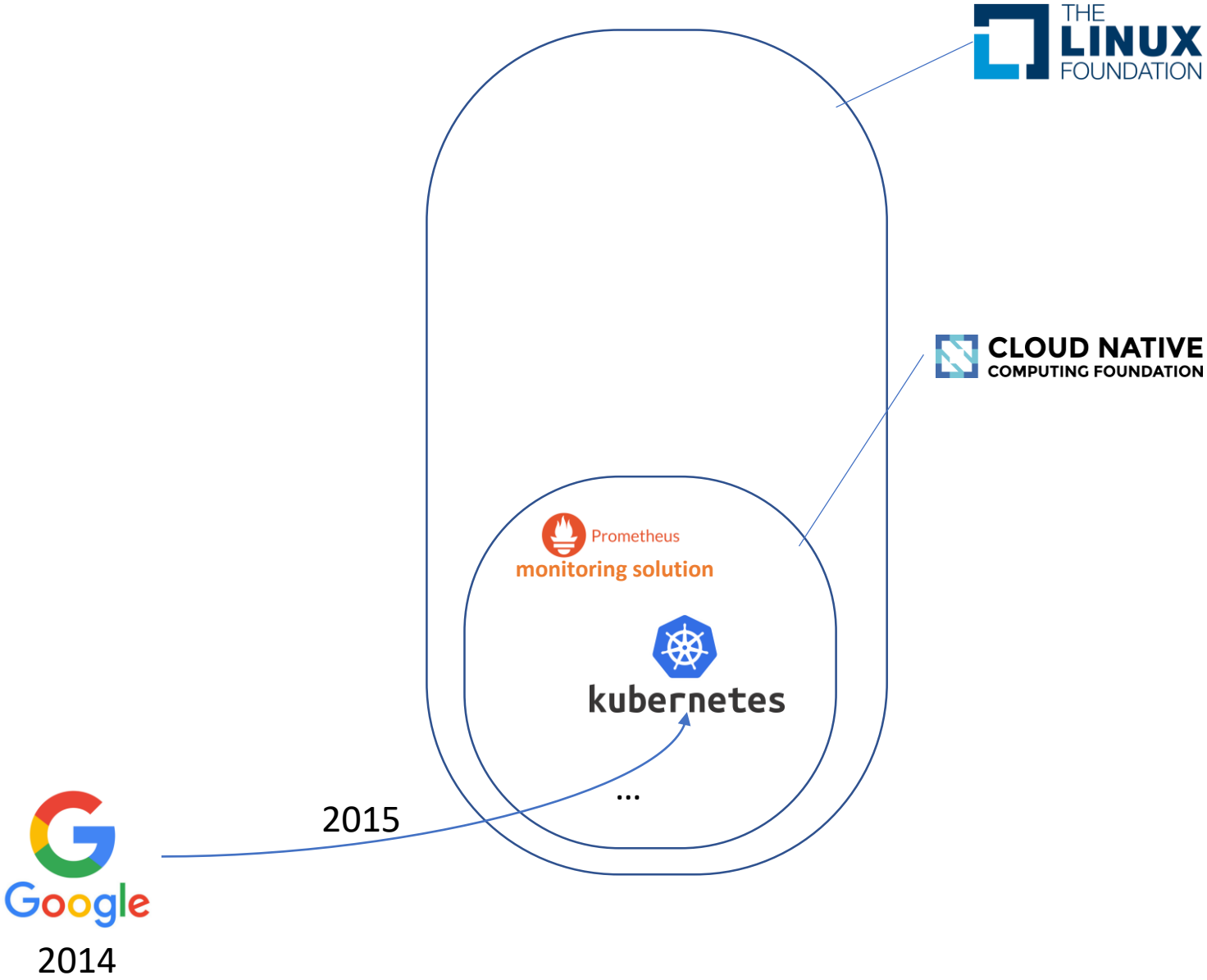


Software Application Engineering

Kubernetes Tutorial

Open-source for the cloud



The Linux Foundation is dedicated to building sustainable ecosystems around open source projects to accelerate technology development and industry adoption […] through financial and intellectual resources, infrastructure, services, events, and training.

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as […] clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

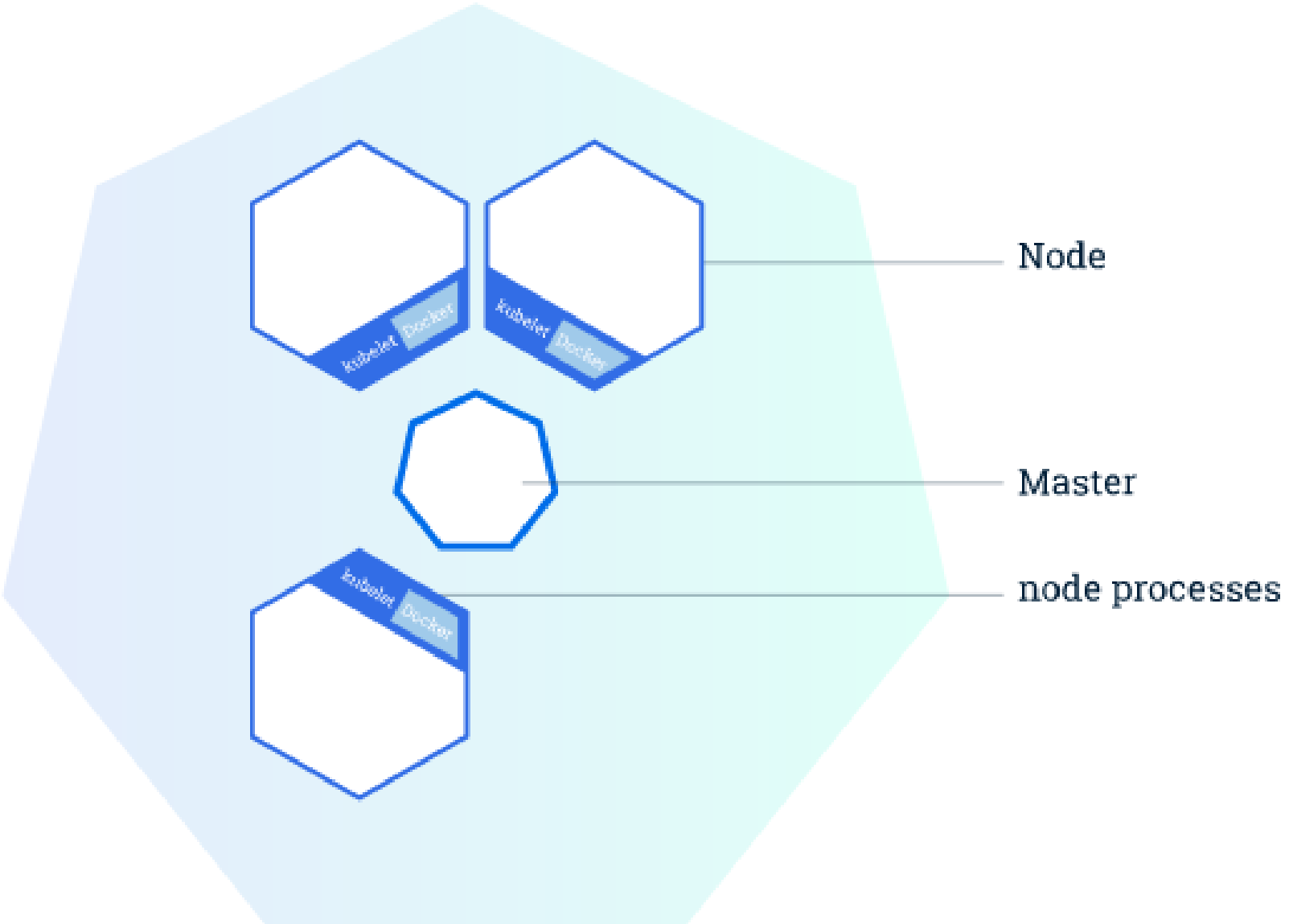
These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

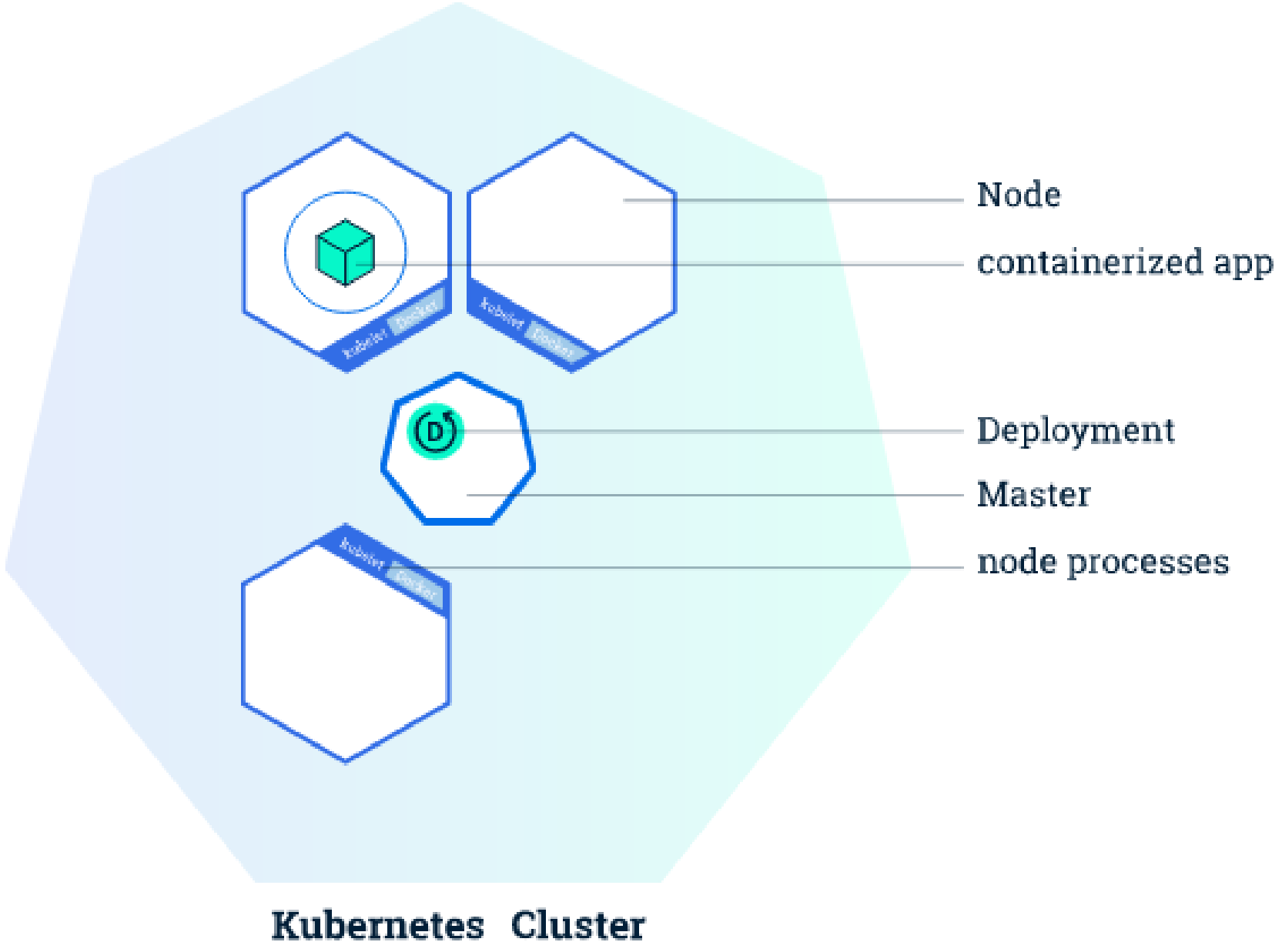
Container orchestration

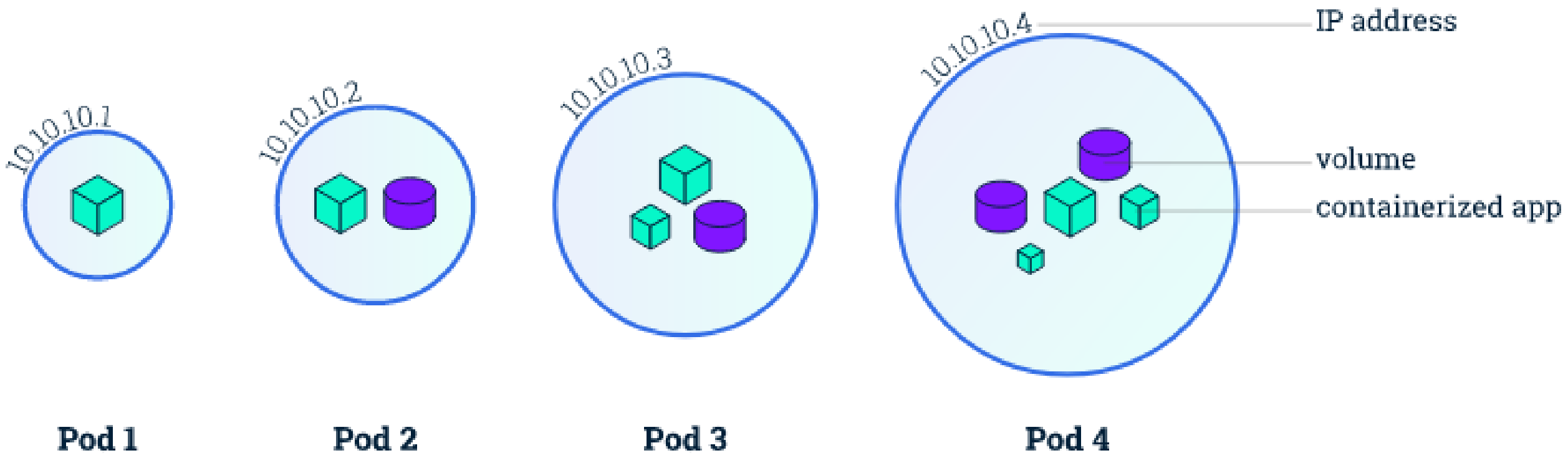
In a nutshell:

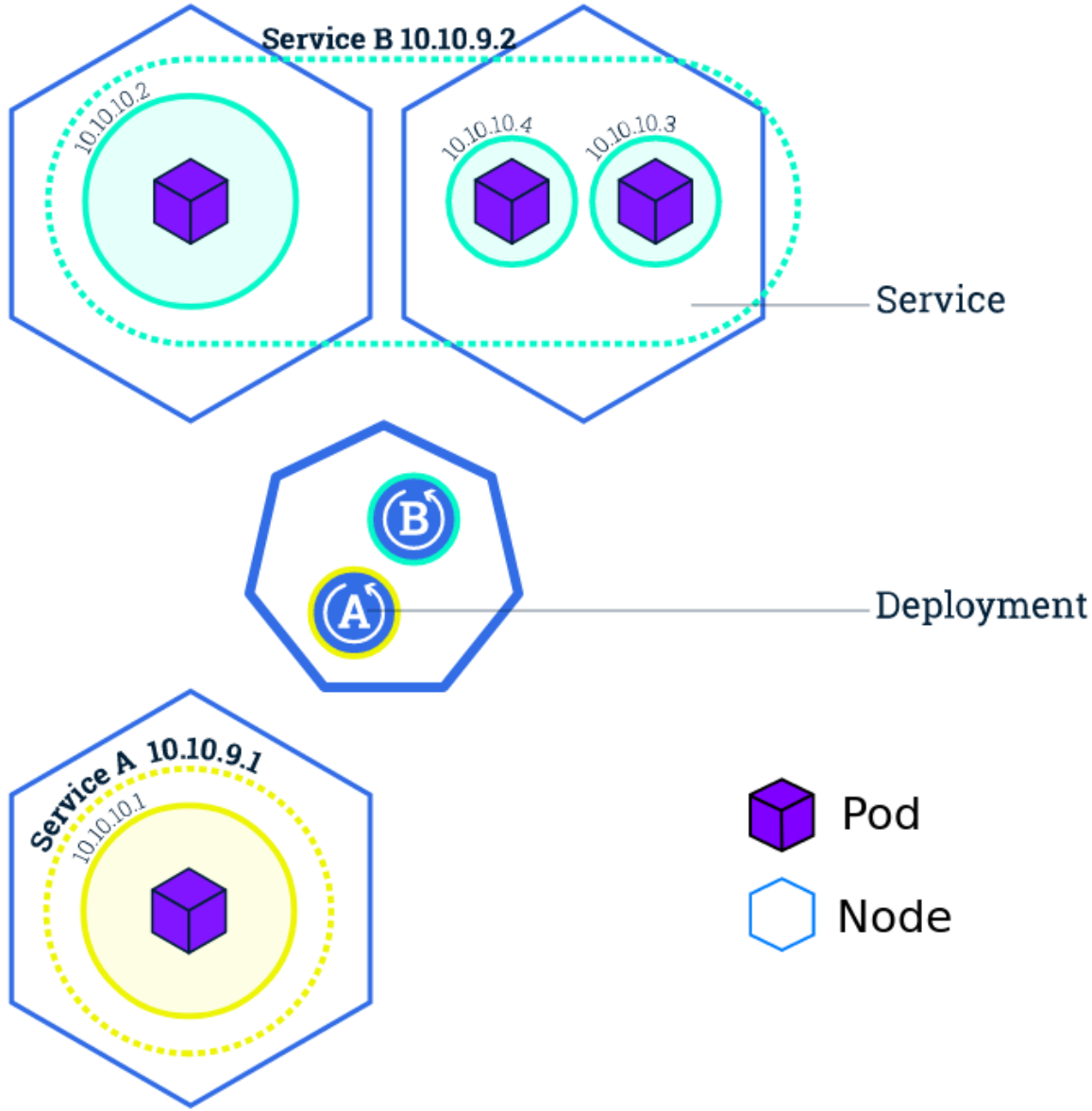
1. write a deployment YAML file that specifies the configuration to run ("desired state"):
   - containers, exposed ports…
   - starting rank…
   - number of replicas
2. submit it to the Kubernetes Cluster Service
3. This service is in charge of managing the cluster of worker nodes to reach and keep the desired state:
   - a worker node hosts containers
   - Is a worker fails, the KCS is responsible for detecting the loss, and reschedule the container(s) somewhere else so that the running configuration matches the desired state



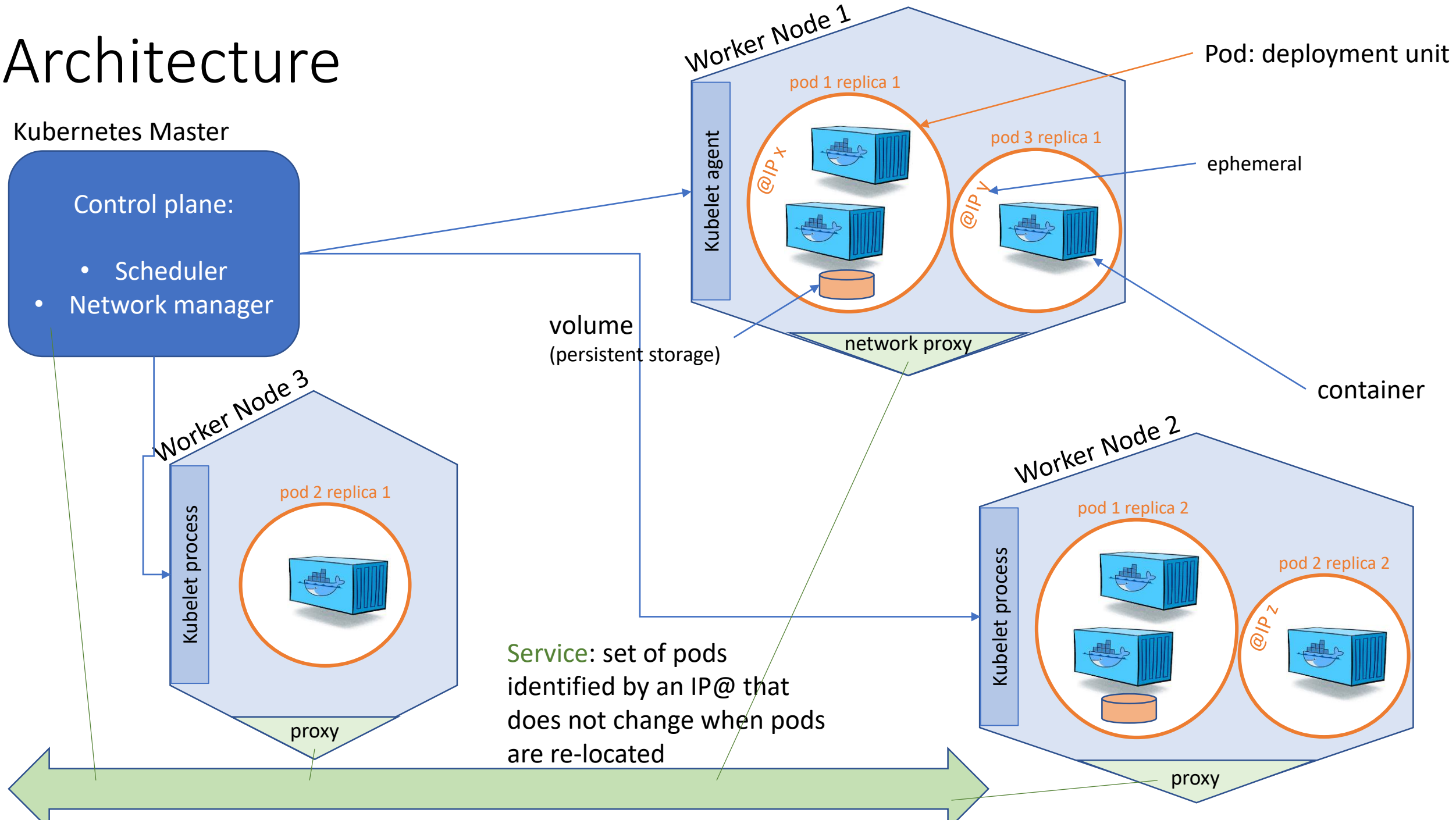
Kubernetes cluster


Node
containerized app
Deployment
Master
node processes
Kubernetes Cluster


10.10.10.1
10.10.10.2
10.10.10.3
10.10.10.4
IP address
volume
containerized app
Pod 1
Pod 2
Pod 3
Pod 4


Service B 10.10.9.2
10.10.10.2
10.10.10.4
10.10.10.3
Service
B
A
Deployment
Service A 10.10.9.1
10.10.10.1
Pod
Node

Architecture



Tutorial Series

minikube

- Tuto#1: Hello World
  - Discover the Minikube Kubernetes cluster
  - Discover the CLI: your first **`kubectl`** commands
  - Run the hello-world Docker image in a pod on a node of Minikube
  - Work with a running pod
  - Transfer files between host and containers with `kubectl cp`
- Tuto#2: on Intercell
  - *namespaces*: one cluster, n isolated environments
- Tuto#3: your first YAML deployment file
- Tuto#4: *hostPath volume* to transfer files between host and pod
- Tuto#5: Networking
  - Within a pod
  - Between pods on the same node
  - Between pods on different nodes
  - Service and *NodePort service*

Slight differences when running minikube within the VM:

- use driver “none”
- sudo most commands
- ssh not possible

Kubernetes workers run on Ic122 to ic133 only

Lab Session (October 20th)

- Deploy your Wikipedia-changes monitor application on Intercell using Docker and Kubernetes
- Demo of kubeview by Patrick Mercier