
ORACLE®

From zero to Kubernetes

Introduction and Overview of Kubernetes

Ioana-Madalina Patrichi <ioana-madalina.patrichi@oracle.com>
Senior Software Engineer
Container Development

Github: https://github.com/MadalinaPatrichi
LinkedIn: https://www.linkedin.com/in/ioana-madalina-patrichi/

June 20, 2018

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

Program Agenda with Highlight

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

What is Kubernetes?

"Kubernetes is an open-source system for automating deployment, scaling and management of containerised applications."

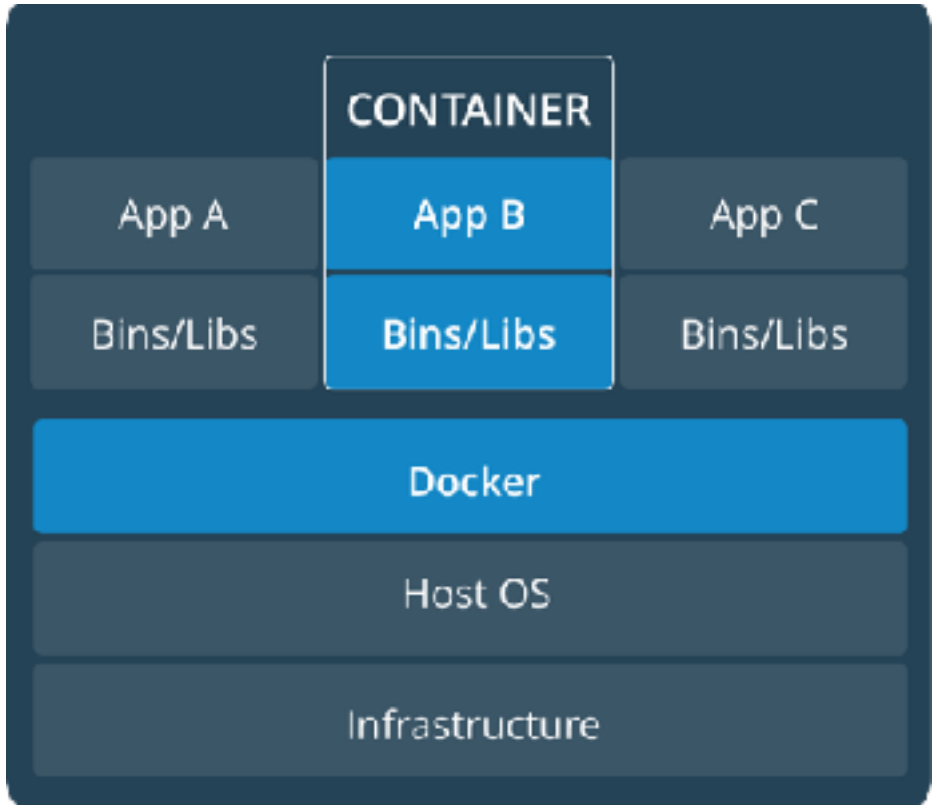
What problems does Kubernetes solve?

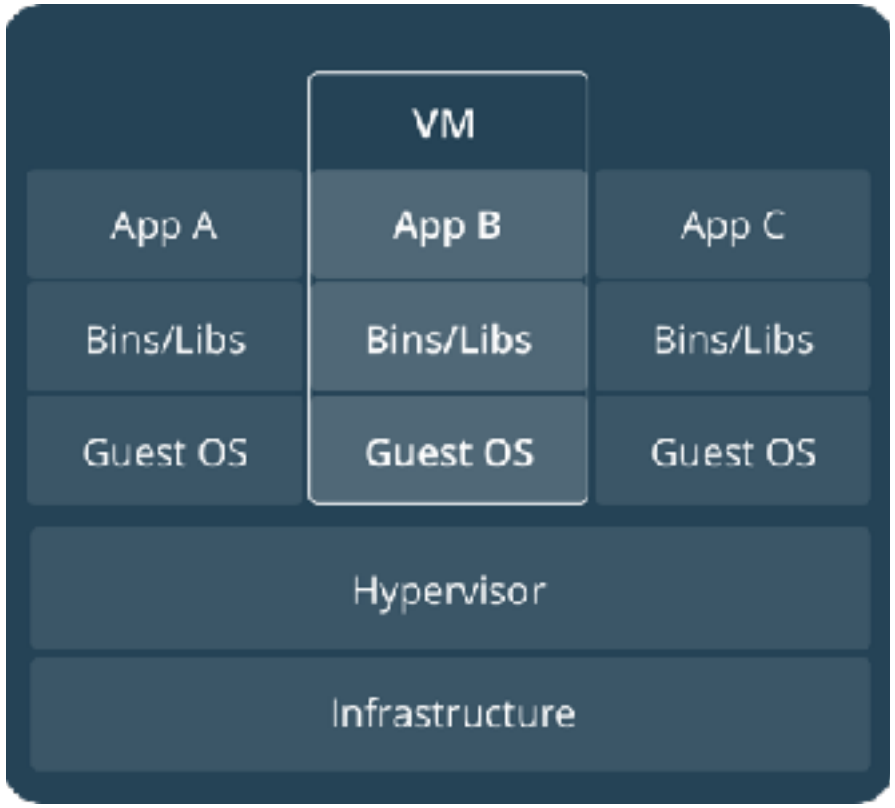
- Users expect applications/services to be available 24/7
- Developers expect to be able to deploy updates to their code multiple times per day
- Desire to use cloud resources efficiently via container orchestration
- Fault-tolerant, self-healing
- Scalability

Program Agenda with Highlight

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

Containers vs VMs





Containers

- Application, not machine centric view of the world
- Containers as a common format for software delivery
- Immutable unit of delivery
- Decouple applications from infrastructure
- Cloud and OS portability

Containers and Container Orchestration

	CONTAINERS	CONTAINER ORCHESTRATION
FUNCTION	Keeps software separated into its own “clean” view of an OS	Defines relationships between containers, where they come from, how they scale and how they connect to the world around them
PREDECESSORS/ALTERNATIVES	• Virtual machines • Direct installation	• Homegrown scripts • Manual, bespoke static configuration between containers
PACKAGES/VENDORS	• Docker • RKT • GARDEN • LXC • Mesos	• Kubernetes • Docker Swarm • Amazon ECS • Mesos

Kubernetes History

- Born from a Google internal project in mid-2014 (Google “Borg”)
- 1.0 release in July 2015
- Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF) to offer Kubernetes as an open standard
- Abbreviated as “k8s”, Greek for “helmsman” or “pilot”
- Written in Golang

Advantages of Kubernetes

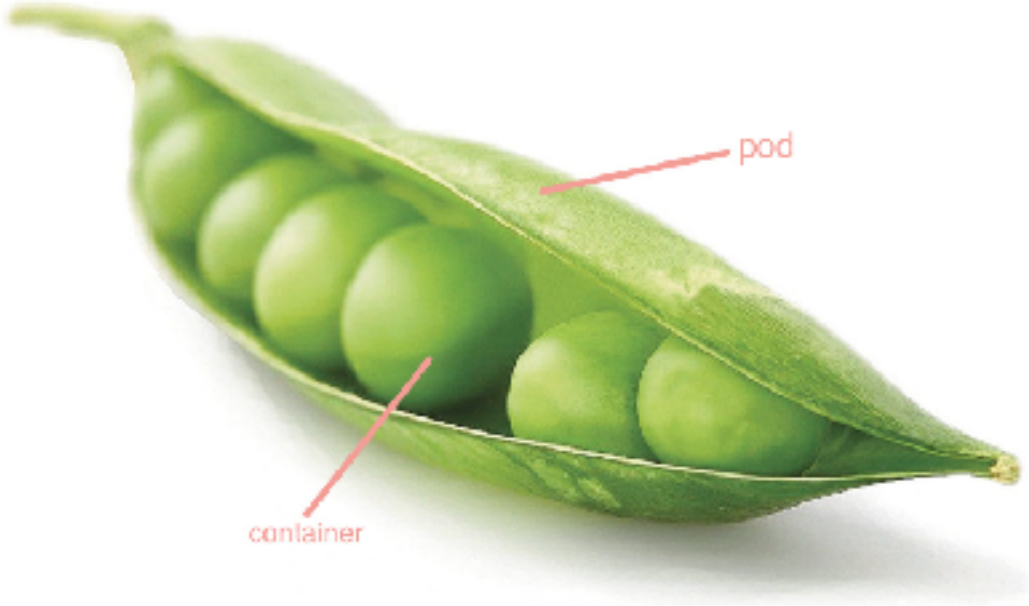
- Based on extensive experience from Google, over a long period of time
- Large open source community &project, mature governing organisation (CNCF)
- Auto-scaling, cloud-agnostic-yet-integratable technologies

Program Agenda with Highlight

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

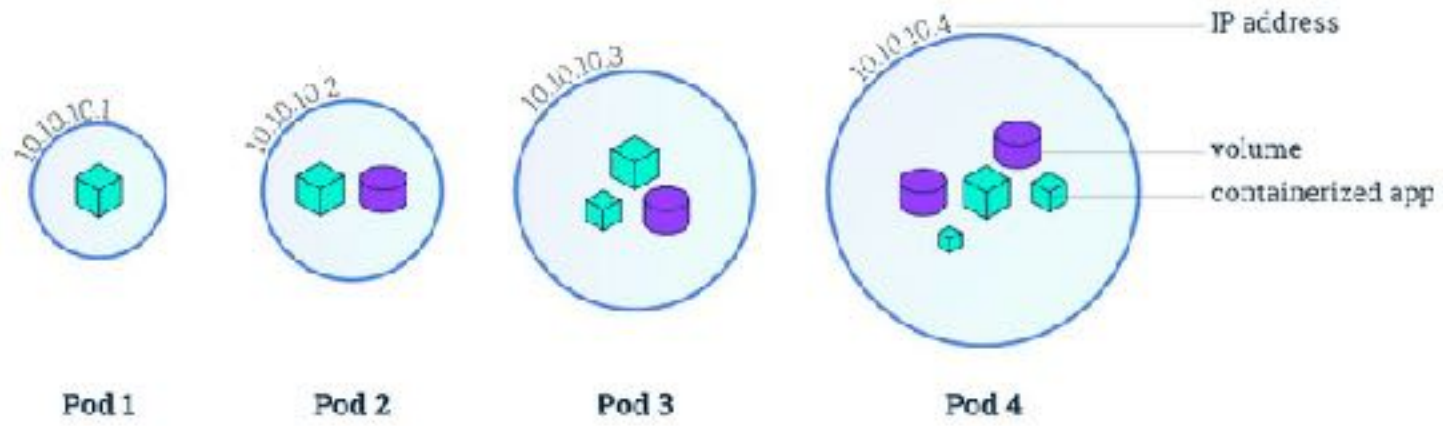
Kubernetes Primitives: Pod

- Set of one or more containers that act as a unit and are scheduled onto a node together
- Share a local network and can share file-system volumes



Kubernetes Primitives: Pod

- Set of one or more containers that act as a unit and are scheduled onto a node together
- Share a local network and can share file-system volumes



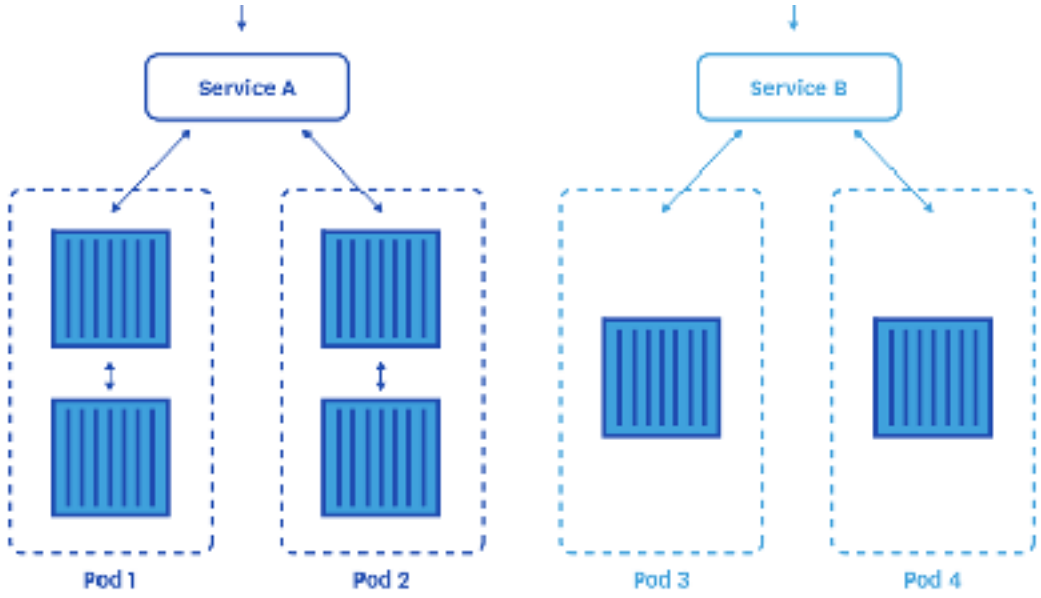
Kubernetes Primitives: Deployment

- Responsible for creating and updating pods
- Kubernetes will manage state based on the definitions provided
- Can scale up/down to meet demand
- Can roll back to older version or roll forward

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 4
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

Kubernetes Primitives: Service

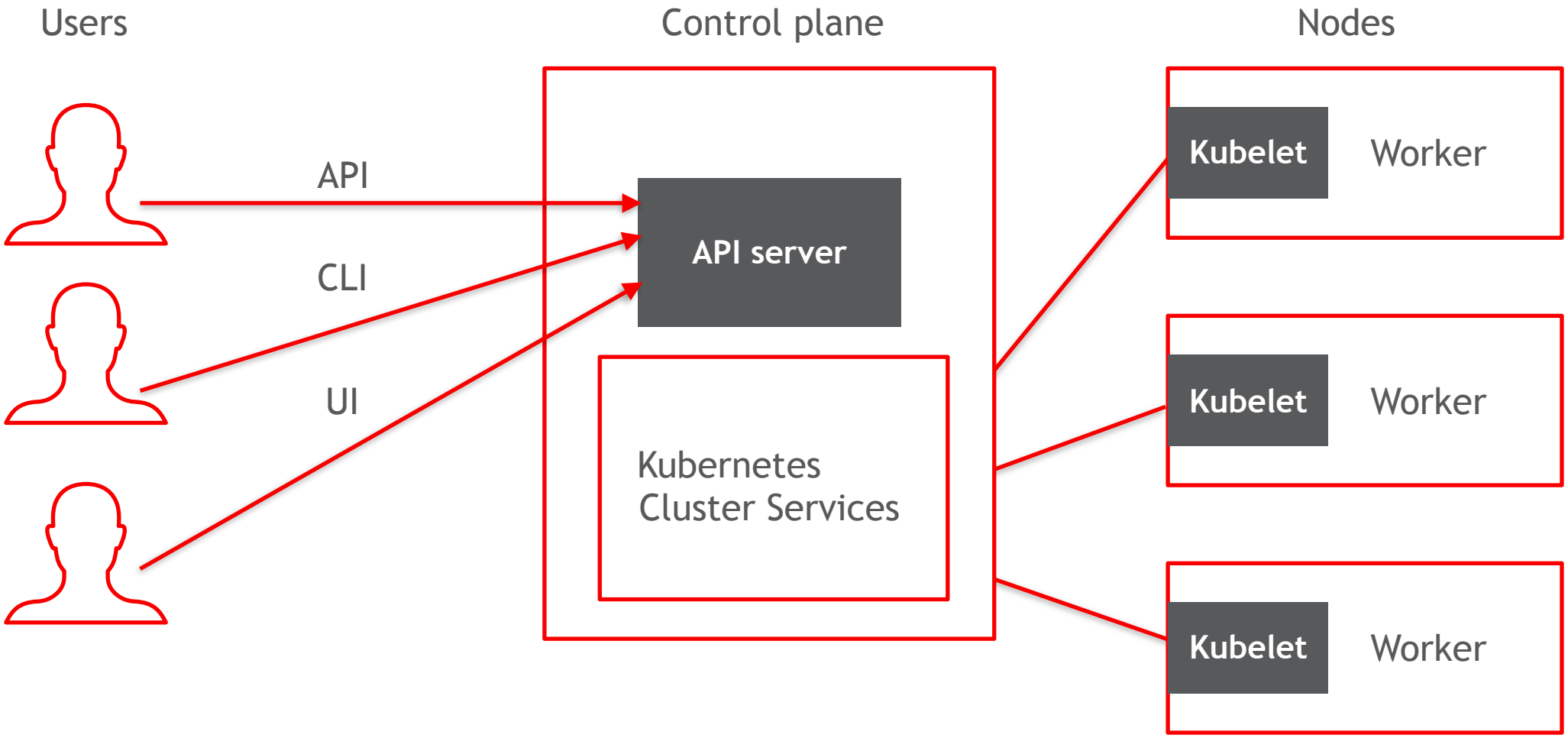
- Defines a way to access pods in a consistent way
- Services find the Pod to route traffic based on the Labels/Selectors in the manifest
  - Inside the cluster, they perform load balancing
  - They can also interact with GKE, OKE, etc to create external load balancers



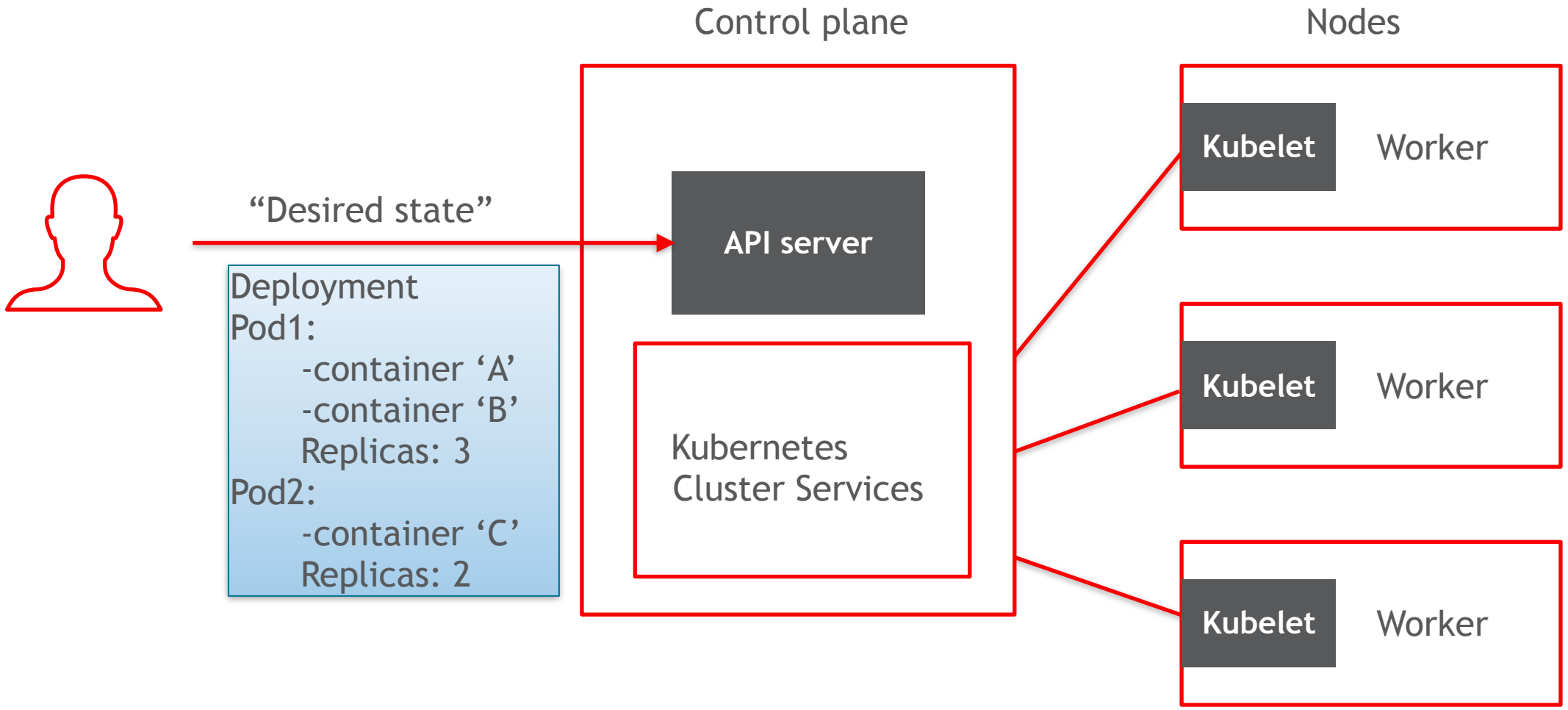
Program Agenda with Highlight

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

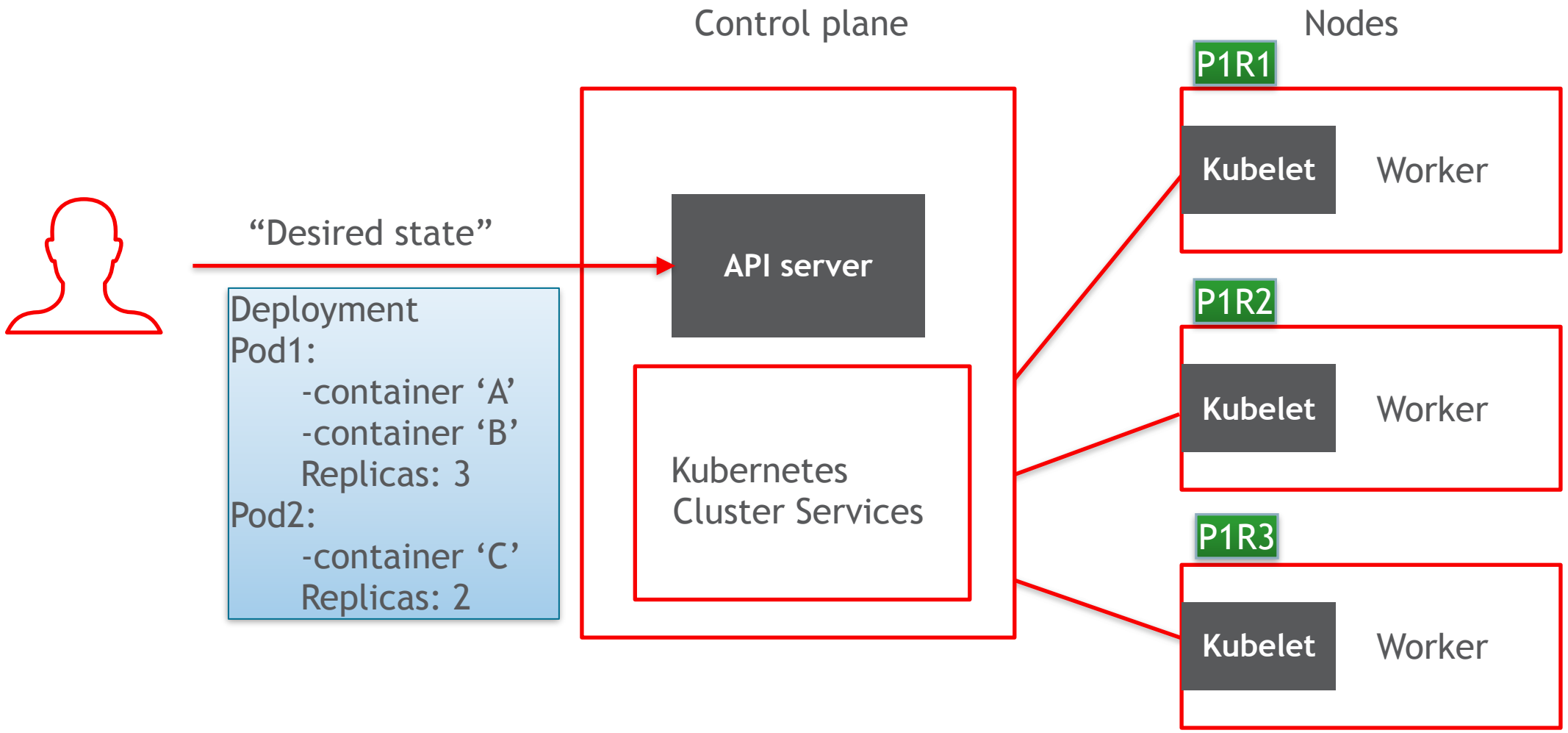
Kubernetes Architecture



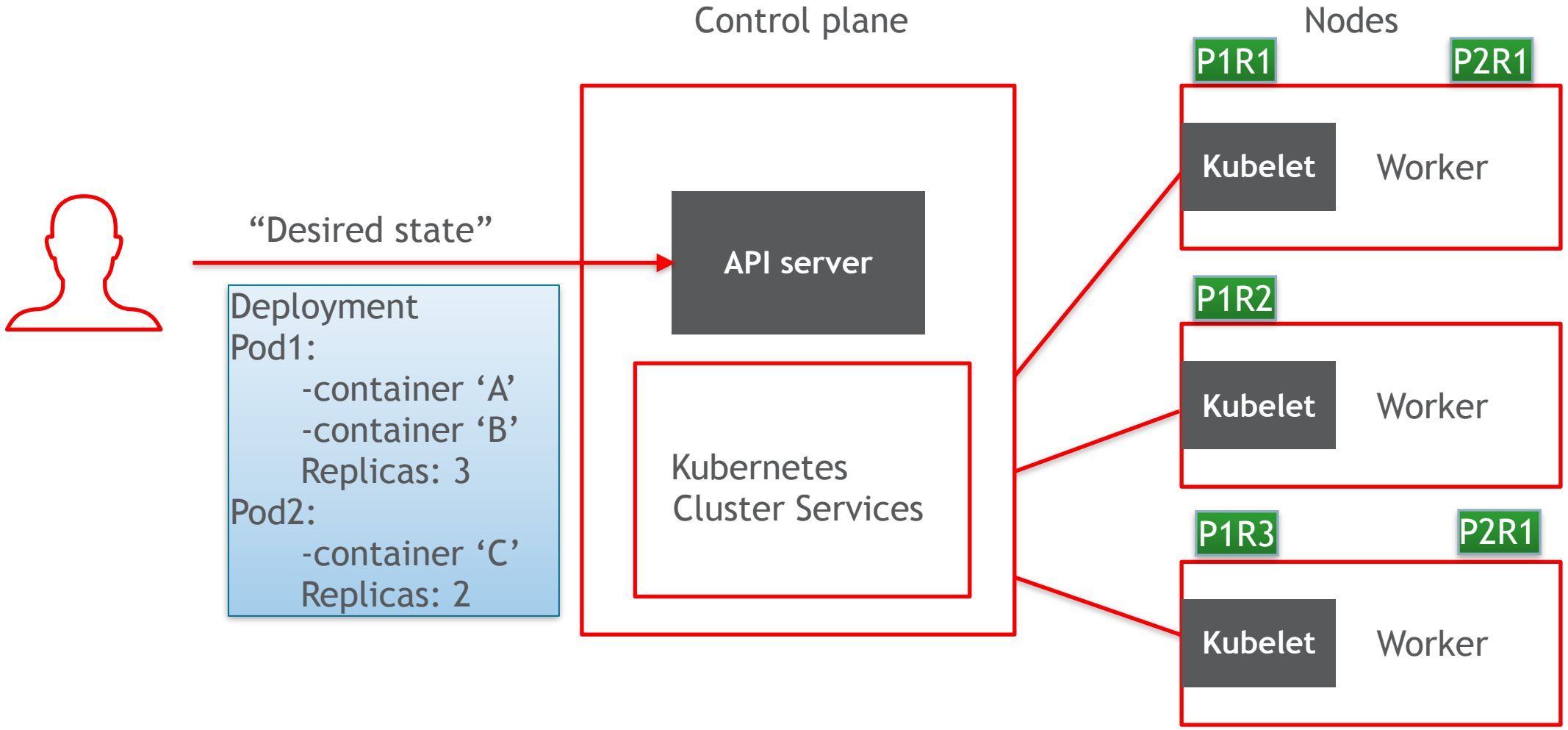
Kubernetes Architecture



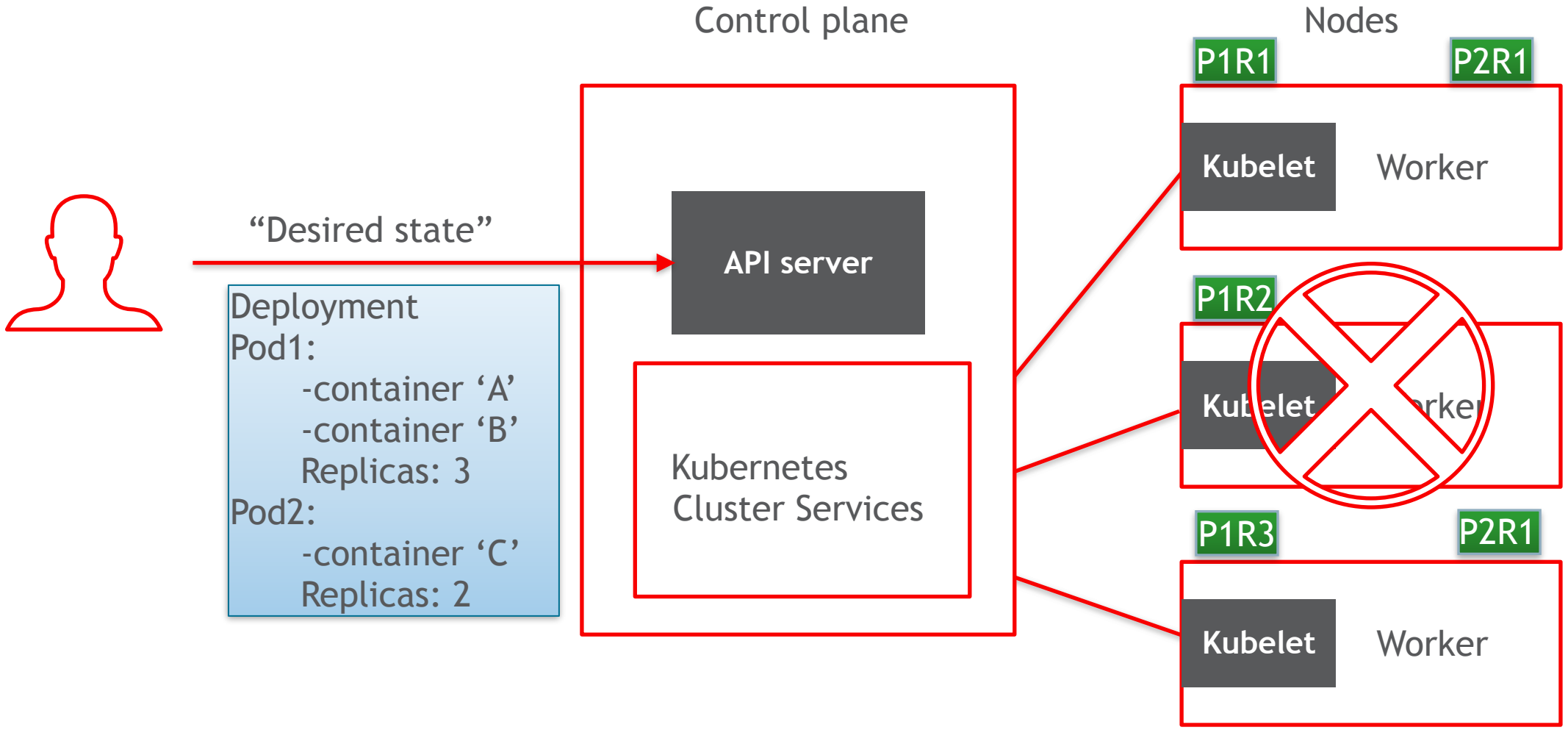
Kubernetes Architecture



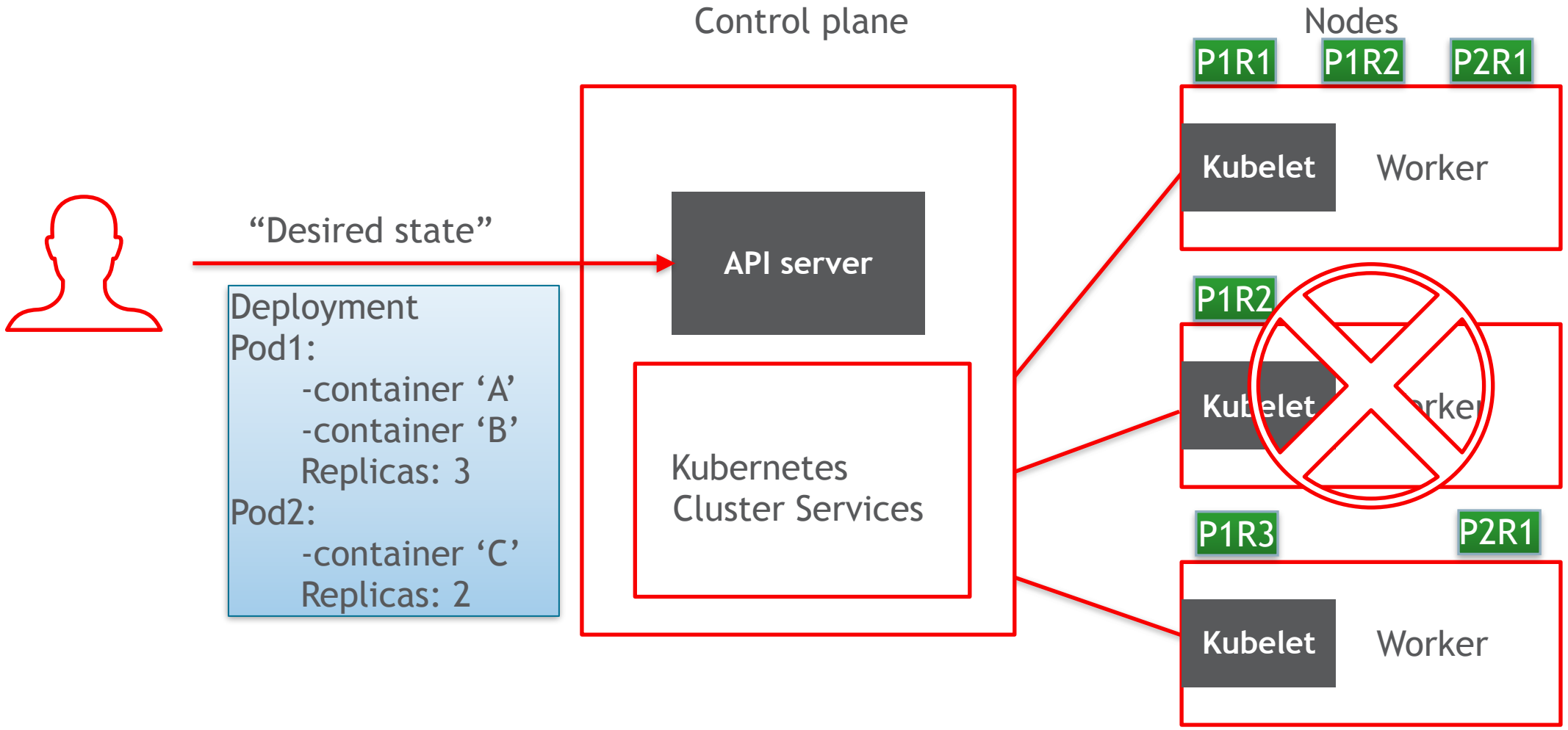
Kubernetes Architecture



Kubernetes Architecture



Kubernetes Architecture



Program Agenda with Highlight

1. Introduction
2. History and overview
3. Terminology
4. Architecture
5. Demo

Useful links:

- Katacoda: https://www.katacoda.com/courses/kubernetes
- Minikube: https://kubernetes.io/docs/tutorials/hello-minikube/
- Oracle Cloud Infrastructure: https://cloud.oracle.com/en_US/tryit
- Terraform: https://www.terraform.io/
- Create your own Kubernetes cluster on OCI: https://github.com/oracle/terraform-kubernetes-installer