Kubernetes

Introduction to Kubernetes

Janak Thakkar (Devops TechLab) (www.devopstechlab.com)

Abstracting Away the Infrastructure

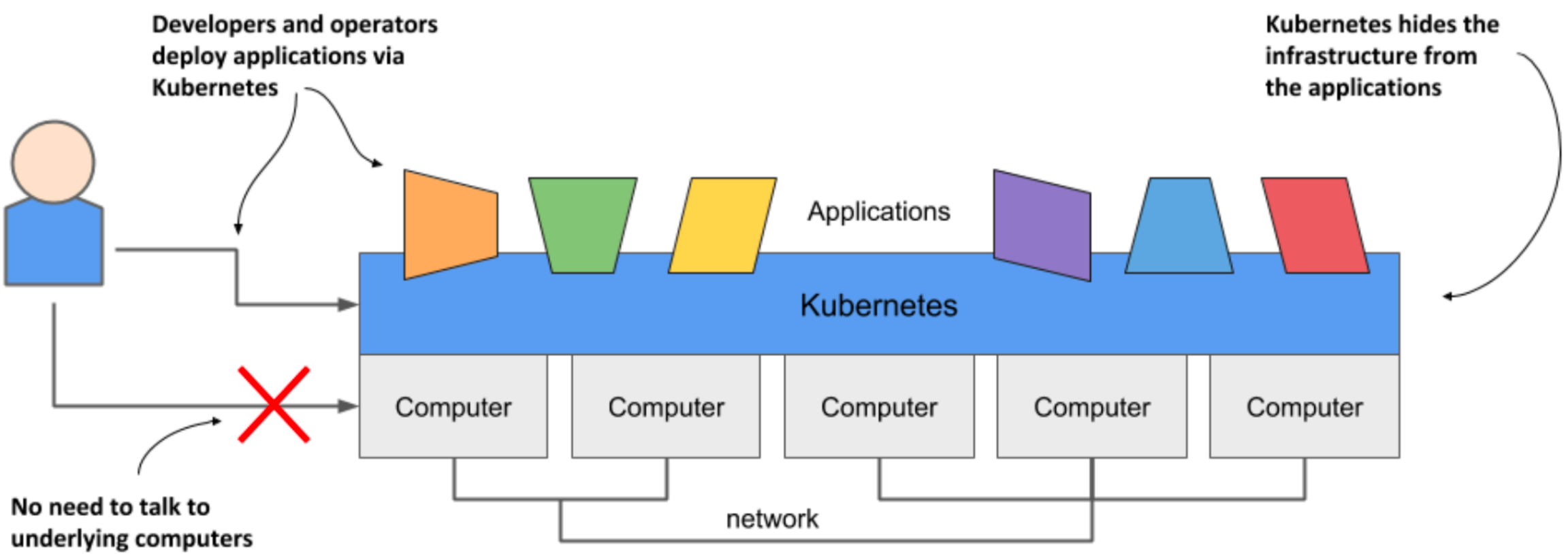
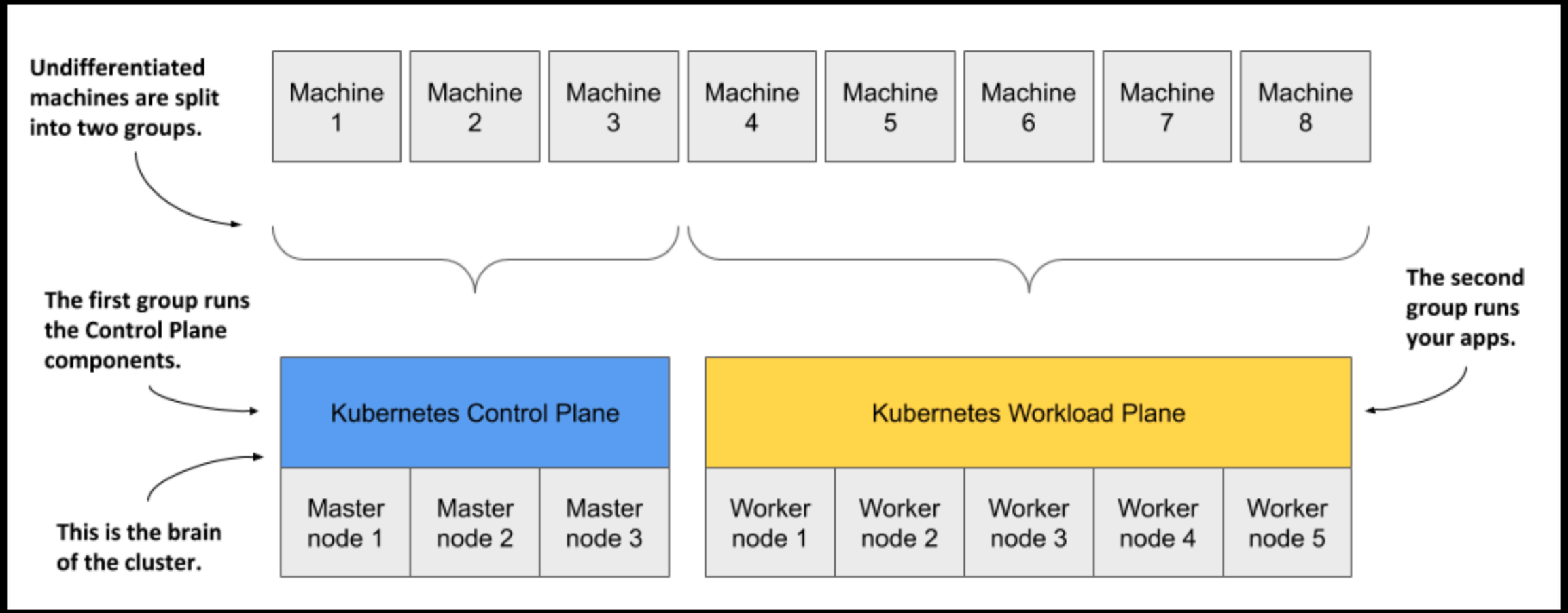


Figure 1.1 Infrastructure abstraction using Kubernetes

How Kubernetes Fits Into a Computer Cluster

Computers in Kubernetes Cluster are divided into Control Plane & Worker Node

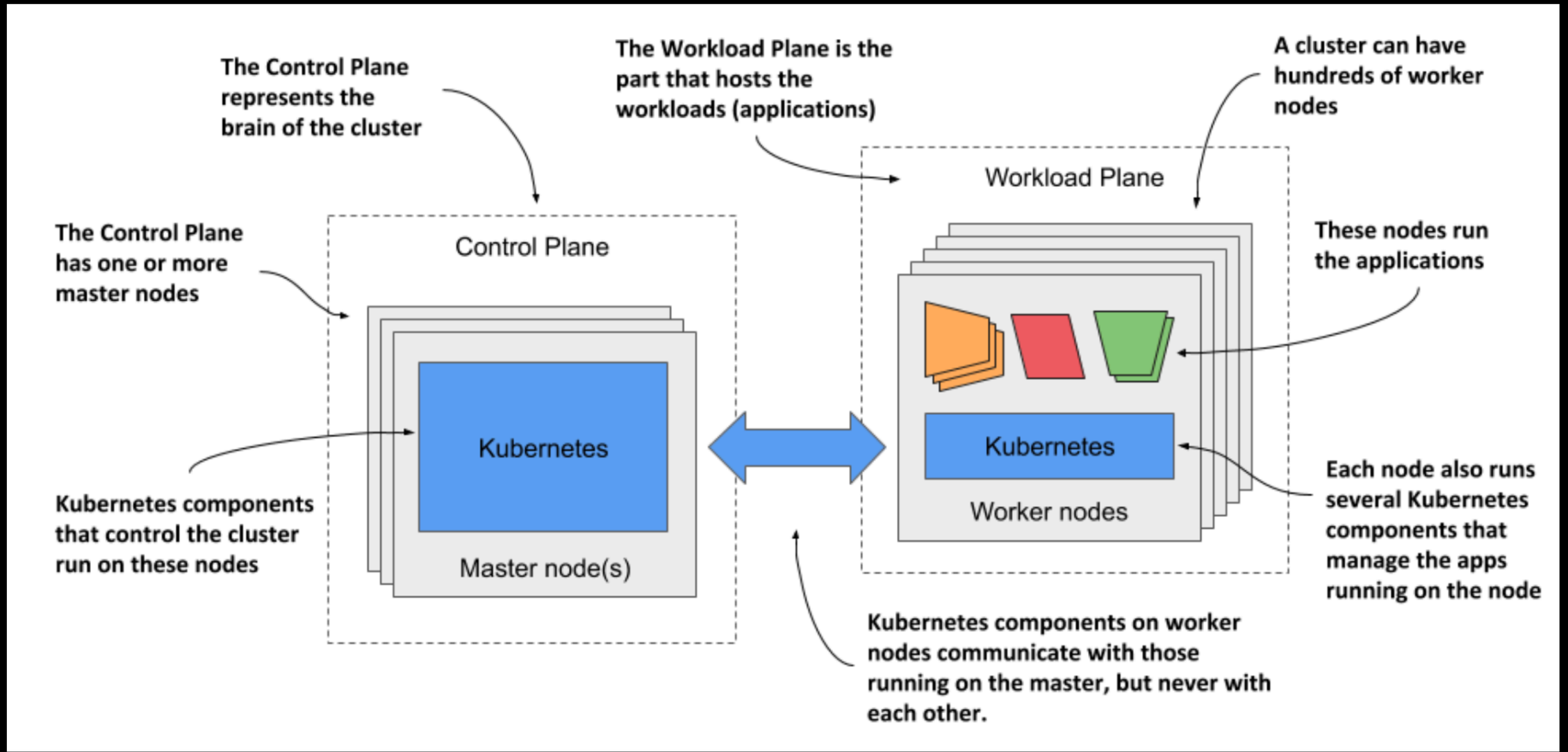


Benefits of Using Kubernetes

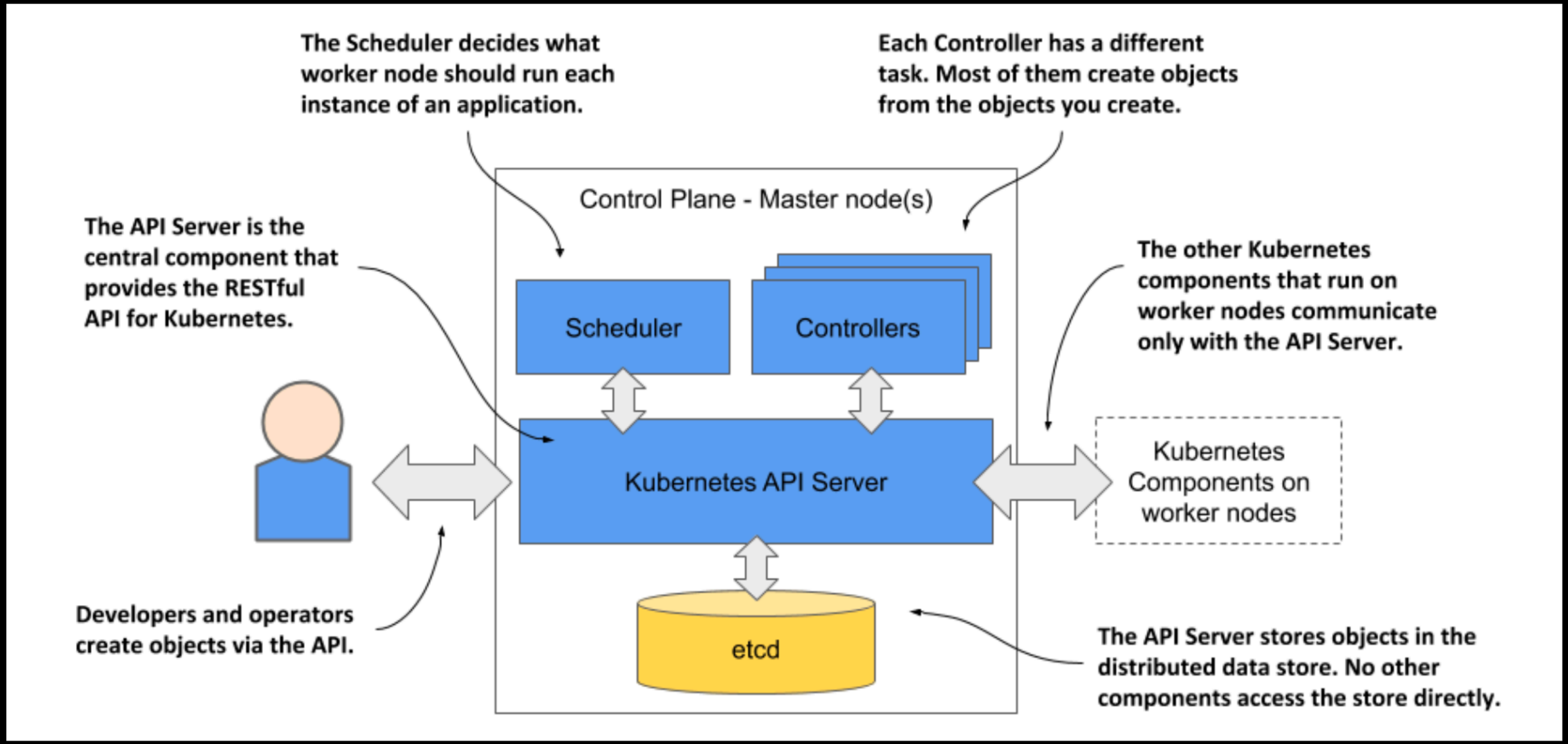
- Self-Service Deployment of Application
- Reducing Cost via better infrastructure utilisation
- Automatically adjusting to changing load
- Keep application running smoothly

Architecture of Kubernetes Cluster

Two planes makes a Kubernetes cluster

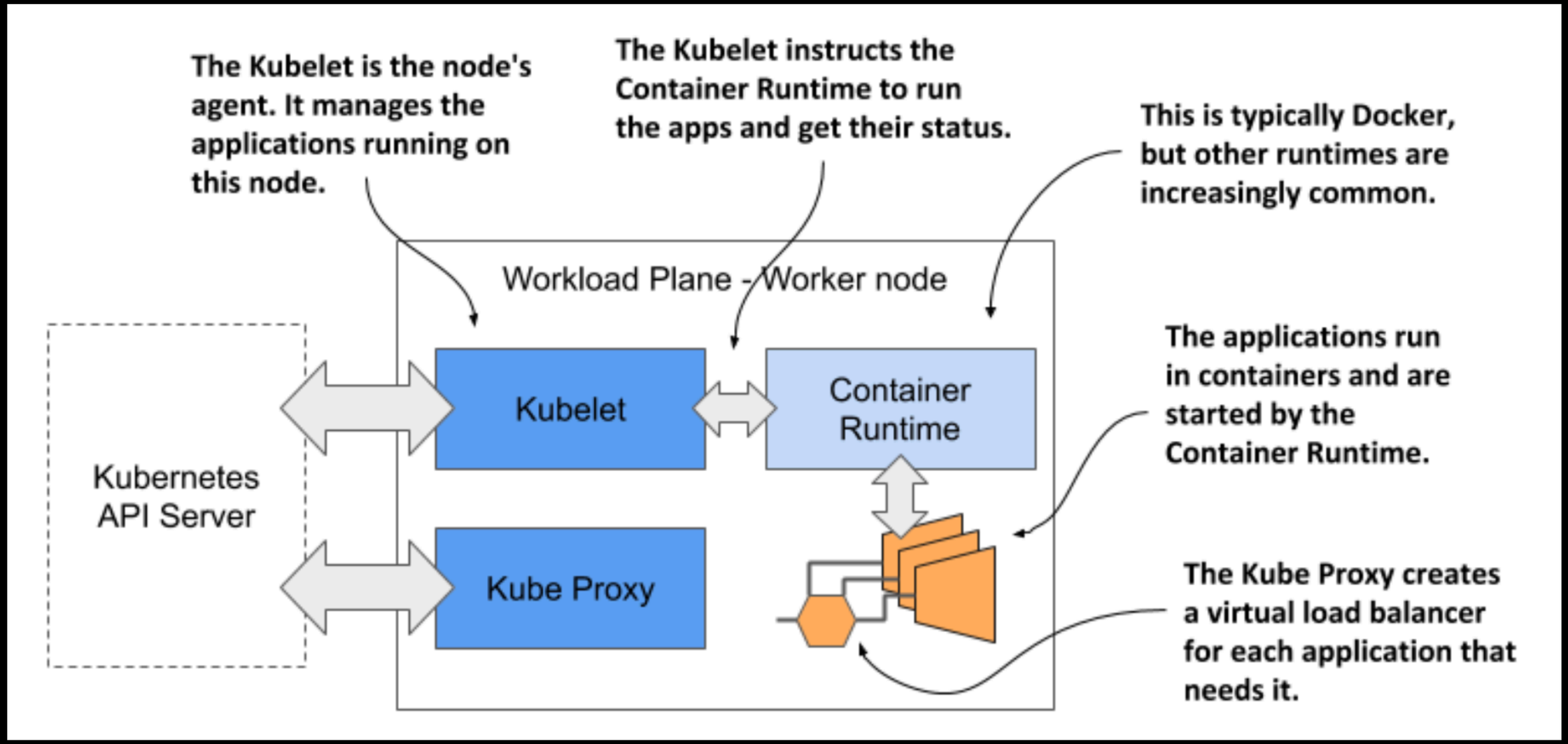


Components of Control Plane



Worker Node Component

Worker Node run your application while control plane hold and control cluster status

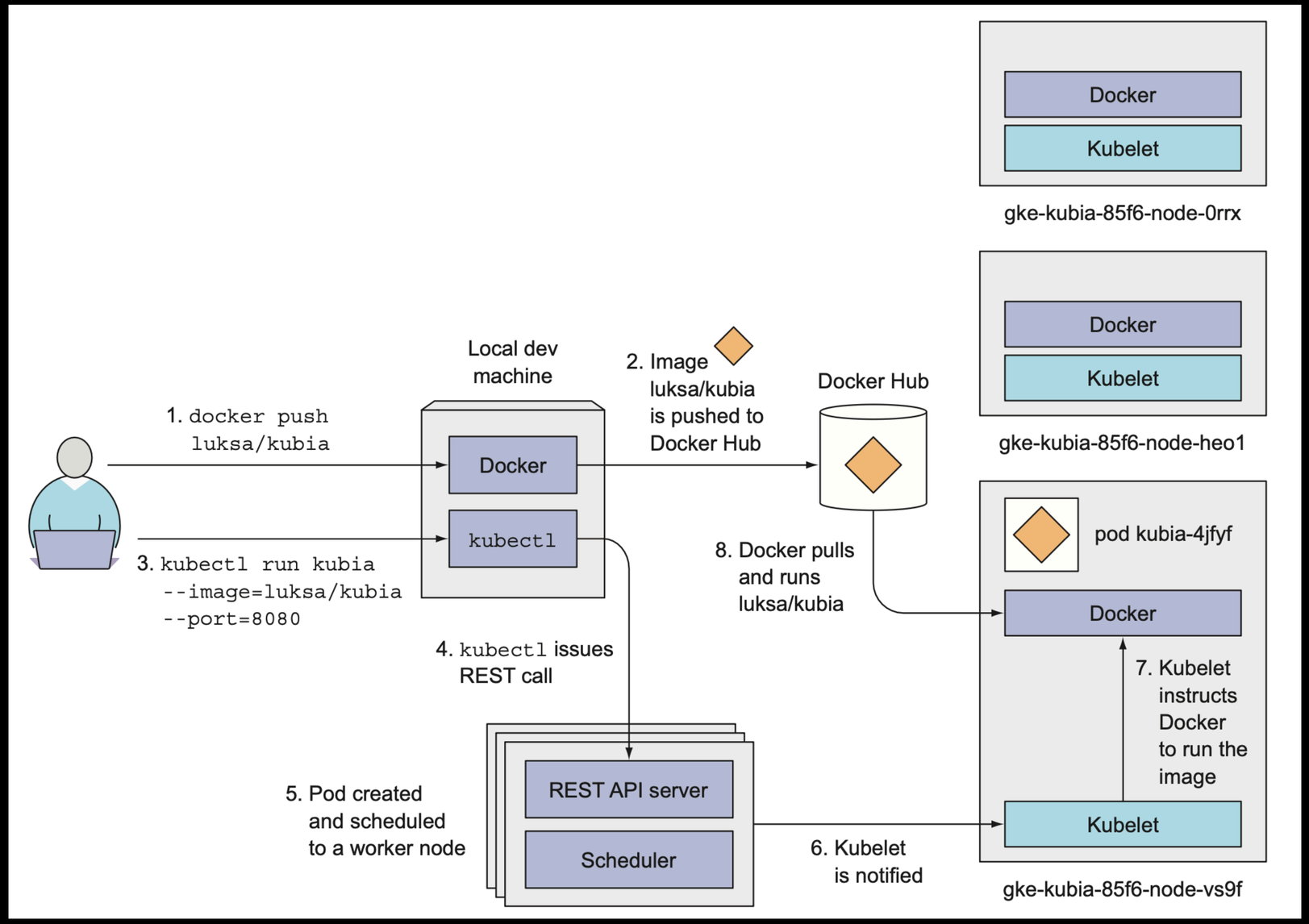


Add-on Components

- Most Kubernetes cluster contains several other components also which run on worker node and some component on master node also.
  - DNS Server
  - Network Pluggins
  - Logging Agents

Introduction to POD

POD Creation - What Happened Behind the Scene



Examining YAML Descriptor of existing pod

```
$ kubectl get po kubia-zxzij -o yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/created-by: ...
  creationTimestamp: 2016-03-18T12:37:50Z
  generateName: kubia-
  labels:
    run: kubia
  name: kubia-zxzij
  namespace: default
  resourceVersion: "294"
  selfLink: /api/v1/namespaces/default/pods/kubia-zxzij
  uid: 3a564dc0-ed06-11e5-ba3b-42010af00004
spec:
  containers:
  - image: luksa/kubia
    imagePullPolicy: IfNotPresent
    name: kubia
    ports:
    - containerPort: 8080
      protocol: TCP
```

Kubernetes API version used in this YAML descriptor

Type of Kubernetes object/resource

Pod metadata (name, labels, annotations, and so on)

Pod specification/ contents (list of pod's containers, volumes, and so on)