



KUBERNETES CONSTRUCTED: IMMERSION IN KUBERNETES

 @MARIOAPARDOO



meetup
Kubernetes Workshop
17th Oct. 2019
Bogotá



kubectl whoami !



 /marioapardo

 @marioapardo0



NAME	GROUPS	ROLES	COMPANY	LOCATION
Mario Pardo	Infra/Cloud	DevOps/SRE	LIFTIT	Bogota, CO
PROFESSION	EXPERIENCE	HOBBIES	UID	
Systems Engineer	+10 years	Photography and Music	@marioapardo	@marioapardo0



Kubectl get agenda!



Session	Name
I	What is Kubernetes?
I	Overview of the different Kubernetes cluster environments
I	Kubernetes Cluster Architecture
II	Immersion in internal components of Kubernetes
II	Running Kubernetes locally with minikube/kind
II	Deploying sample application on kubernetes
III	Secrets, volumes and annotations with k8s resources.
III	Introduction to ingress controllers and ingress routing.
III	Q/A & Networking



What is Kubernetes?



- Container-Orchestration
- Portable
- Extensible
- Open-Source
- Written in Go
- Released by Google to the community in mid-2014 and based on the Borg product.
- K8S → K(ubernetes)
- Greek word Helsman
- 80000+ commits
- 2300+ contributors
- 900+ groups meetup and 400+ of meetups around the world



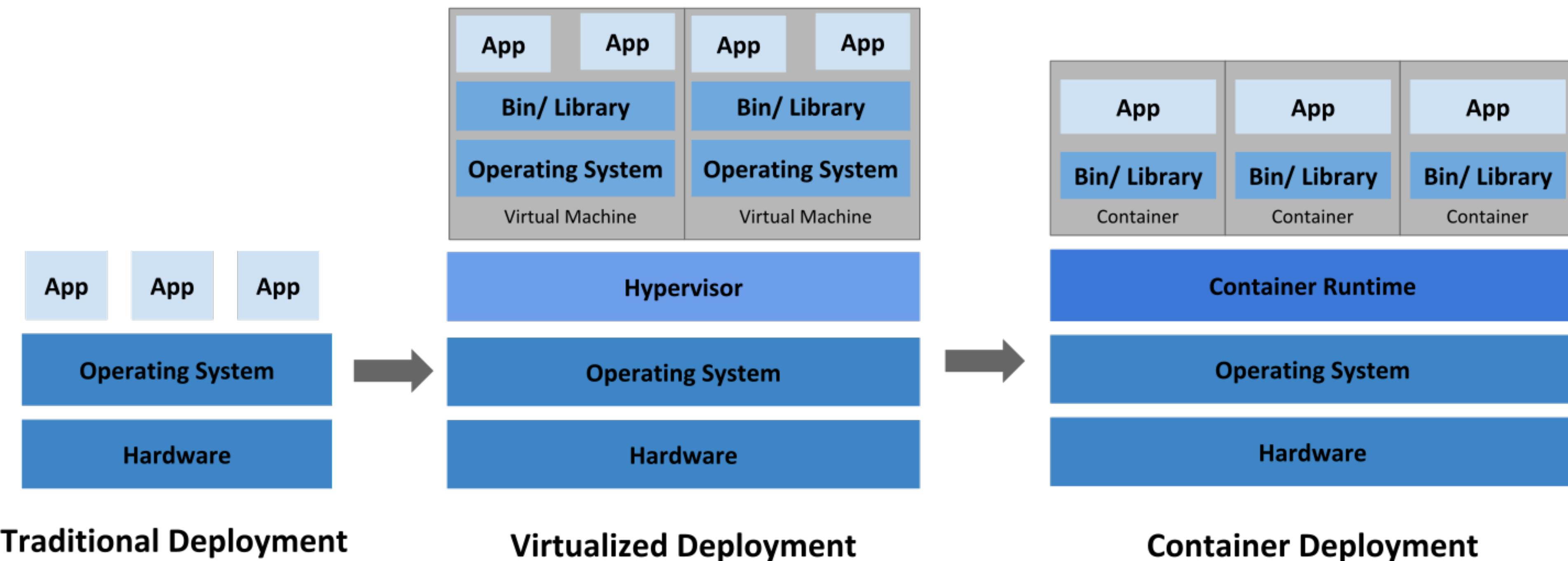
Pre-History



- 1979 chroot.
- 2000 FreeBSD Jail.
- 2005 Solaris Zones/containers.
- 2005 OpenVZ.
- 2006 Linux Cgroups and Namespaces/Net
- 2006 Cloud Provider VMs (AWS)
- 2008 LXC
- 2010 OpenStack
- 2013 Docker
- 2014 Kubernetes



Pre-History





Overview of the different Kubernetes cluster environments



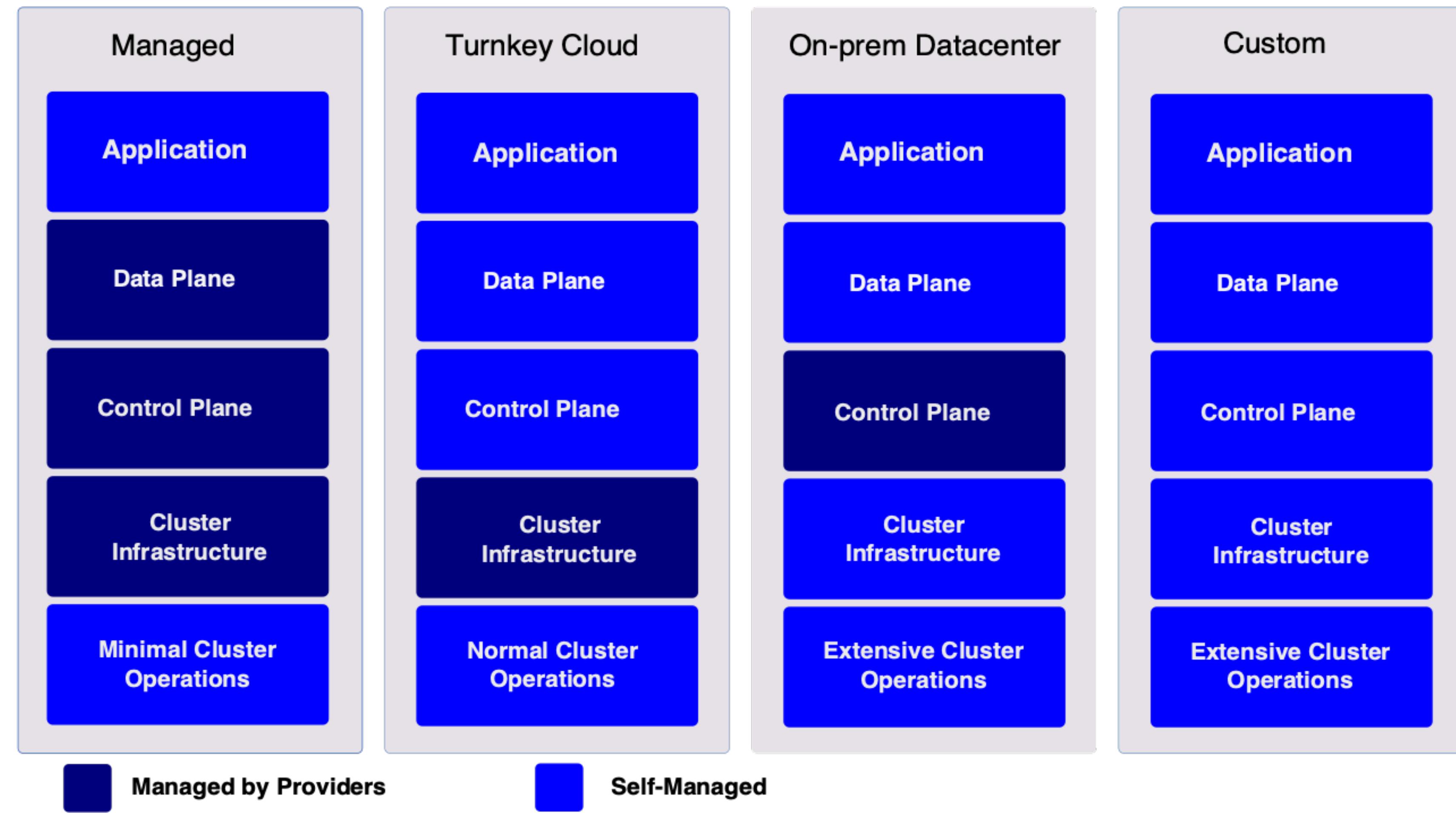
- Learning environment.
- Production environment.



Overview of the different Kubernetes cluster environments



- Production environment.





Kubernetes Cluster Architecture



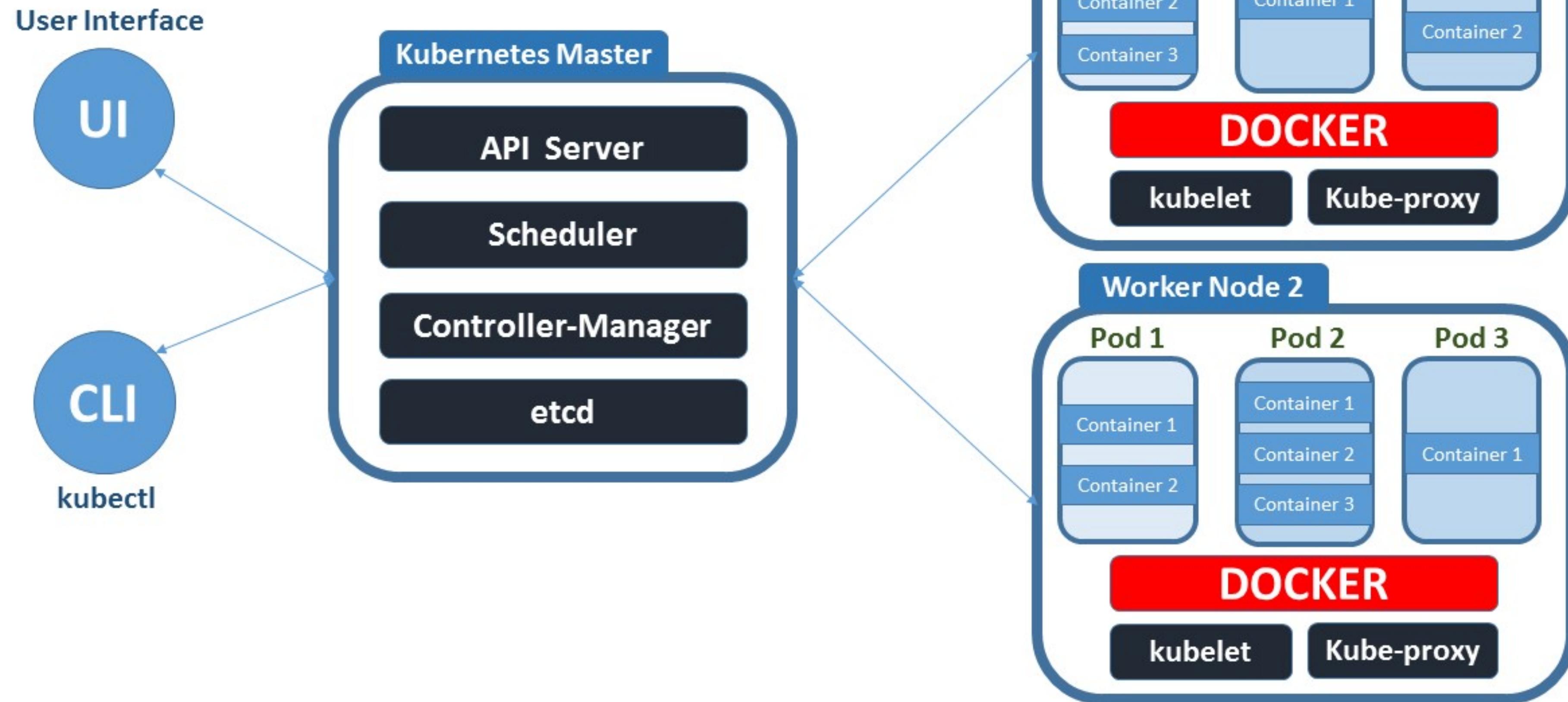
- Master Node
 - API-Server
 - Scheduler
 - Controller
 - ETCD
- Node (minion)
 - Kubelet
 - KubeProxy
 - ContainerRuntime



Kubernetes Cluster Architecture



KUBERNETES ARCHITECTURE

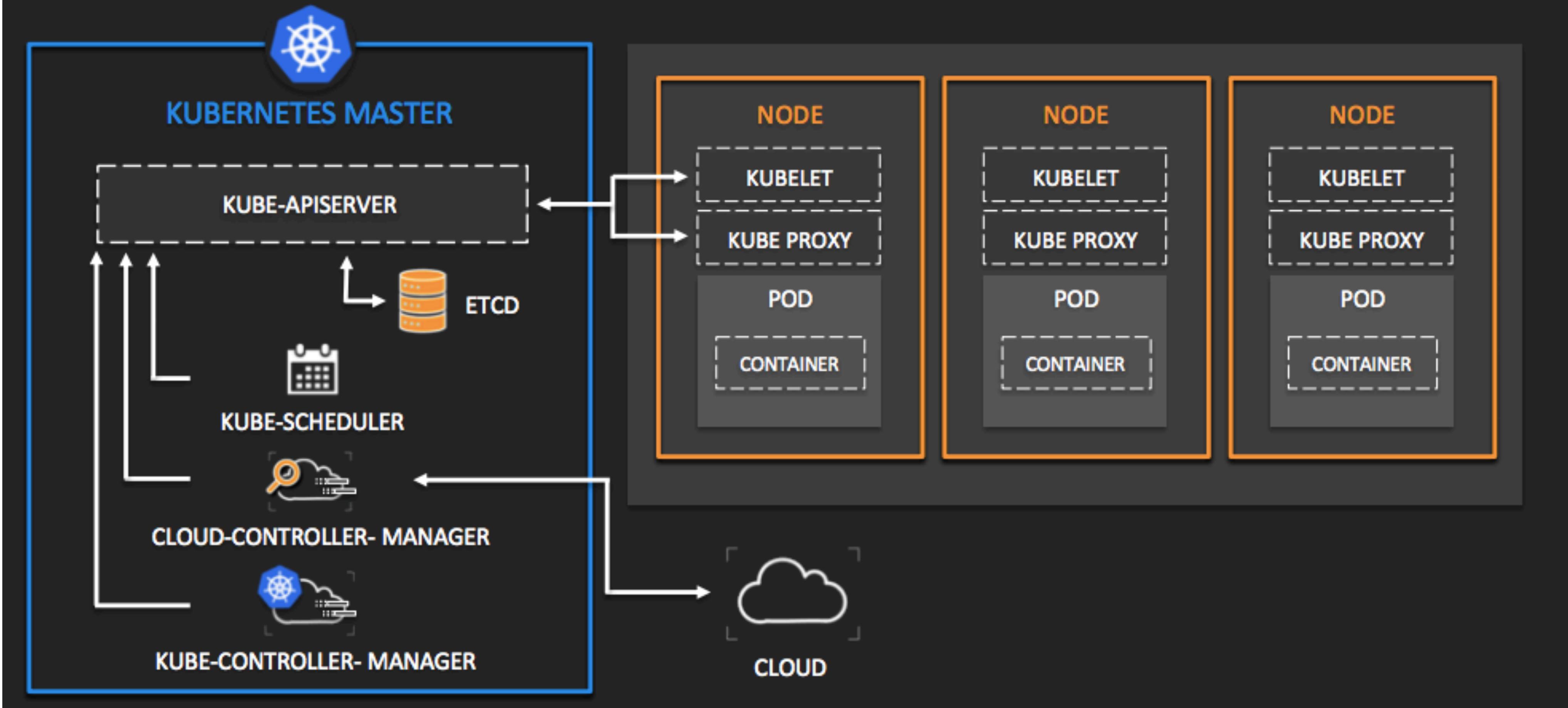




Kubernetes Cluster Architecture



BASIC KUBERNETES ARCHITECTURE





Resources

- Workloads
 - Pods
 - Containers
 - Init Containers
 - Controllers
 - ReplicaSet
 - ReplicaControllers
 - Deployment
 - StatefulSet
 - DaemonSet
 - Jobs
 - Cronjobs
- Services and Networking
 - Services
 - ClusterIP
 - NodePort
 - LoadBalancer
 - ExternalName
 - Ingress
 - Nginx
 - HA-Proxy
 - Istio/Envoy
 - Traefik
 - ..



Resources

- Configuration
 - ConfigMaps
 - Secrets
 - Labels
- Storage
 - Volumes
 - PersistentVolumes
 - StorageClasses
 - Dynamic Volume Provisioning
- Objects
 - Namespaces
 - Labels and selectors
 - Annotations, etc ..



K8S - Ready



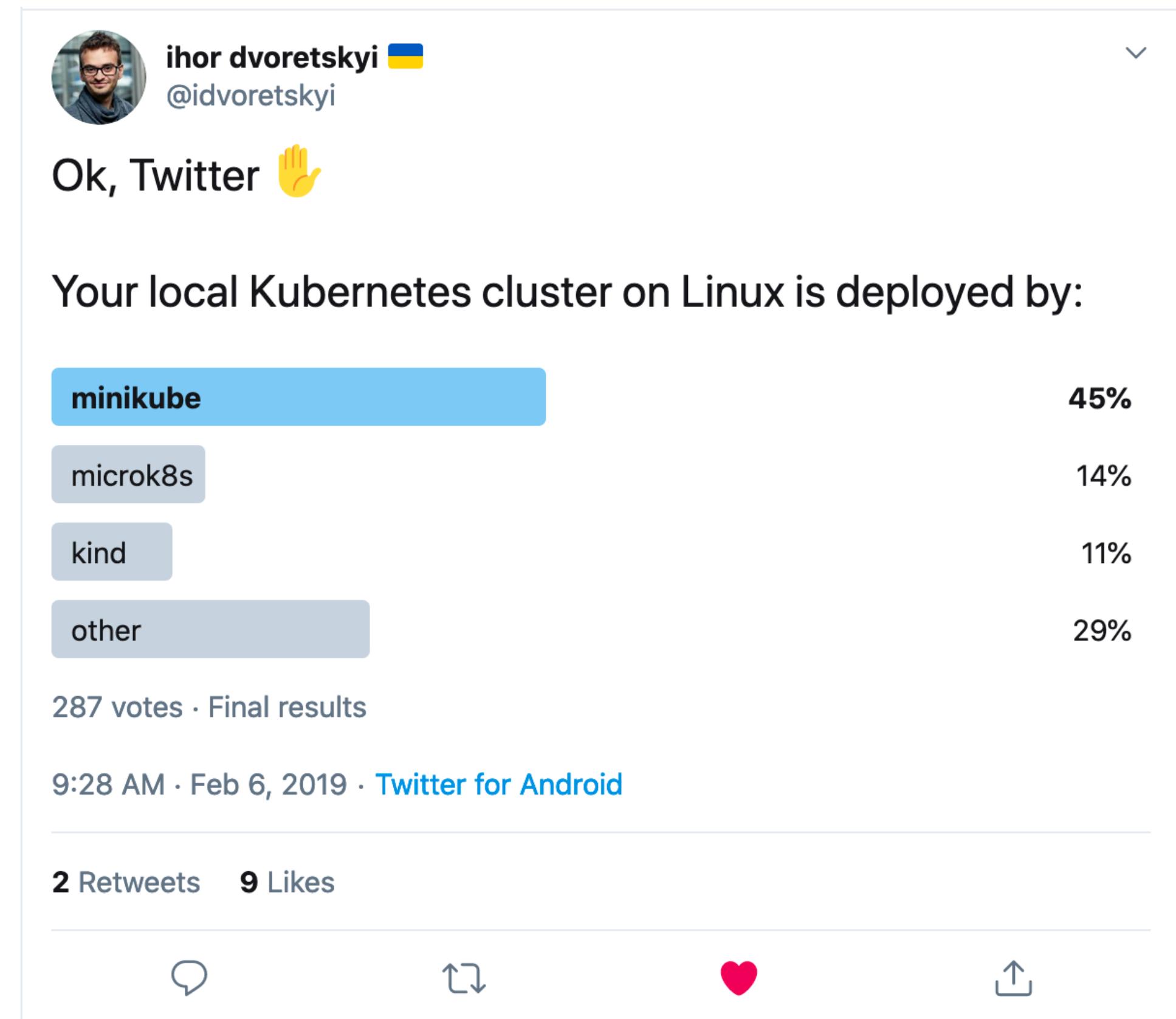
DEMO



Running Kubernetes locally



- Minikube
- Kind (Kubernetes IN Docker)
- K3S (Rancher)
- Docker Desktop (Win, macOS)
- MicroK8s (Ubuntu)(Multipass on Win/macOS)
- Kubeadm (in single-node mode!)





Running Kubernetes with Minikube



Prerequisites

- Win/macOS/Linux
- VMware/Virtualbox installed and properly configured
- The Kubernetes CLI (kubectl) for operating the Kubernetes cluster



Running Kubernetes with Minikube



Minikube

```
→ minikube start --vm-driver=vmware --cpus=2 --memory=2g
😄 minikube v1.4.0 on Darwin 10.15
🔥 Creating vmware VM (CPUs=2, Memory=2000MB, Disk=20000MB) ...
🌐 Preparing Kubernetes v1.16.0 on Docker 18.09.9 ...
🚜 Pulling images ...
🚀 Launching Kubernetes ...
⏳ Waiting for: apiserver proxy etcd scheduler controller dns
🏄 Done! kubectl is now configured to use "minikube"
```

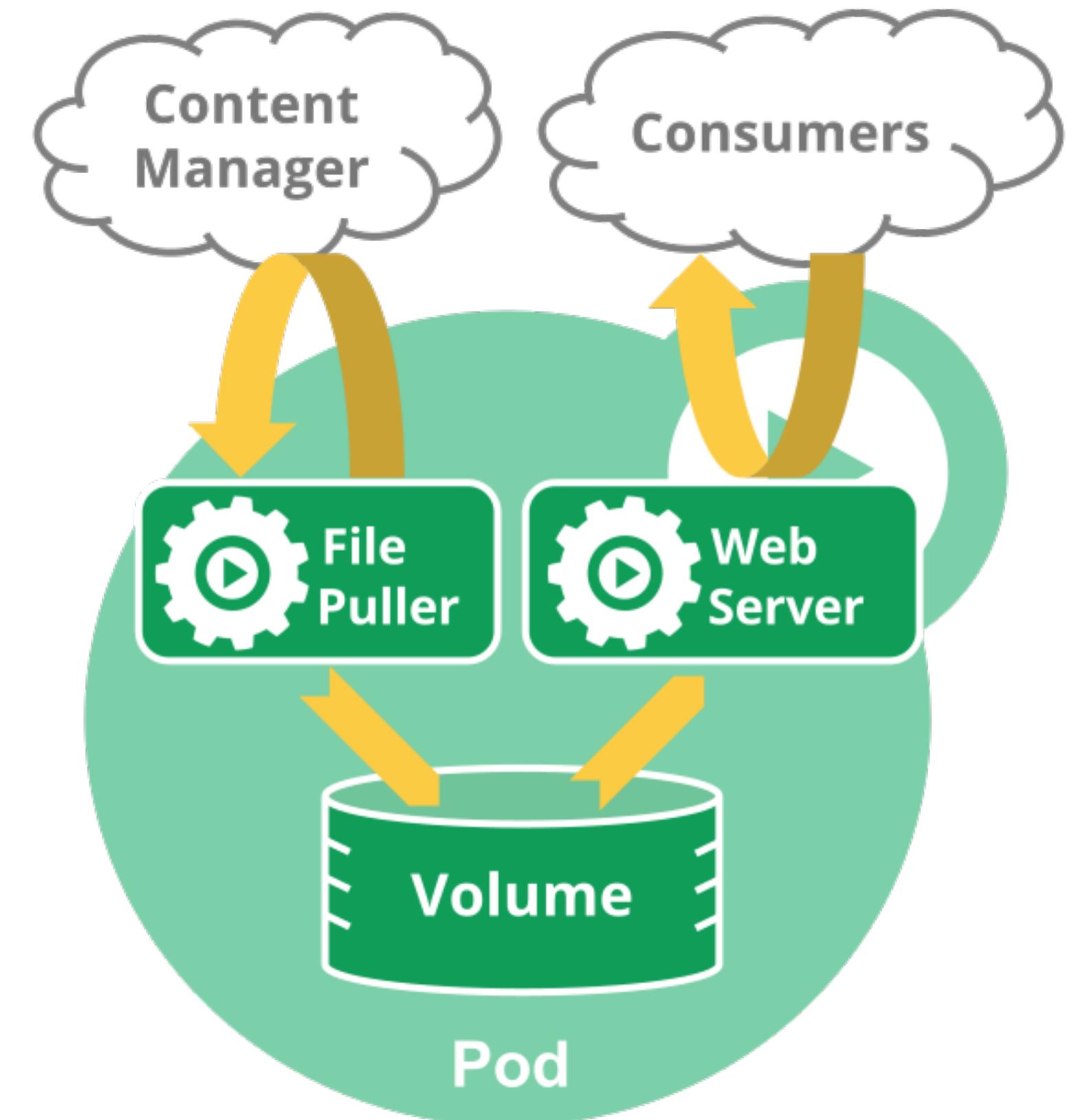


Deploying sample application on kubernetes



Pods

- A pod (as in a pod of whales or pea pod) is a group of **one or more containers**
- Containers within a pod share an **IP address** and port space, and can find each other via **localhost**
- Containers in a Pod also share the same data **volumes**
- Pods are considered to be **ephemeral**





Labels

- A Label is a **key/value** pair attached to Pods and convey user-defined attributes.
- You can then use label selectors to select Pods with particular Labels and apply **Services** or **Replication Controllers** to them.
- Labels can be attached to objects at creation time and subsequently added and modified at any time



Deploying sample application on kubernetes



Pod →

nginx.yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

labels:

name: nginx-pod

spec:

containers:

- name: nginx

image: nginx:latest

ports:

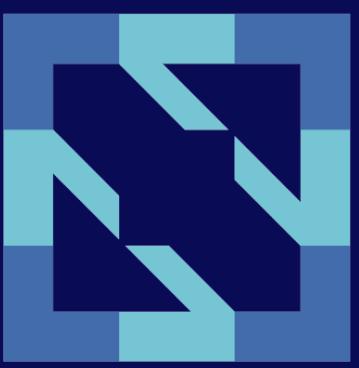
- containerPort: 80

```
→ kubectl create -f nginx-pod.yml  
pod/nginx-pod created
```

```
→ kubectl get pod nginx-pod  
NAME      READY   STATUS    RESTARTS   AGE  
nginx-pod  1/1     Running   0          27s
```



Deploying sample application on kubernetes



Services

Kubernetes provides several ways to expose services to the outside:

- **NodePort:** with this method Kubernetes exposes the service through special ports (30000-32767) of the node IP address.

- **Loadbalancer:** with this method Kubernetes interacts with the cloud provider to create a load balancer that redirects external traffic to the Pods

- **Ingress Controller:** Since Kubernetes v1.2.0 it's possible to use Kubernetes ingress which includes support for TLS and L7 http-based traffic routing



Deploying sample application on kubernetes



Service → **service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  selector:
    name: nginx-pod
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 80
    type: NodePort
```

```
→ kubectl create -f nginx-service.yml
service/nginx created

→ kubectl get service
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1   <none>       443/TCP     19h
nginx      NodePort   10.111.0.58  <none>       80:32031/TCP 36s

→ minikube service nginx
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default   | nginx | http        | http://172.16.242.132:32031 |
|-----|-----|-----|-----|
🌟 Opening kubernetes service default/nginx in default browser...
```