DEPLOYMENT ORCHESTRATION

G. Molines

2020-2021

Université Nice Sophia Antipolis

POLYTECH NICE-SOPHIA

DOCKER COMPOSE

Features

Assemble several containers in one go

→Virtual network

→Shared resources, Esp. volume

→Build and Run

Limits

Runs on one host

Scaling

Monitoring

Resiliency

DOCKER SWARM

Swarm

Scale containers

Now native in docker engine

Easy to start with, fast

Not really picking up in the industry

Nice tutorial: https://training.play-with-docker.com/orchestration-hol/

KUBERNETES

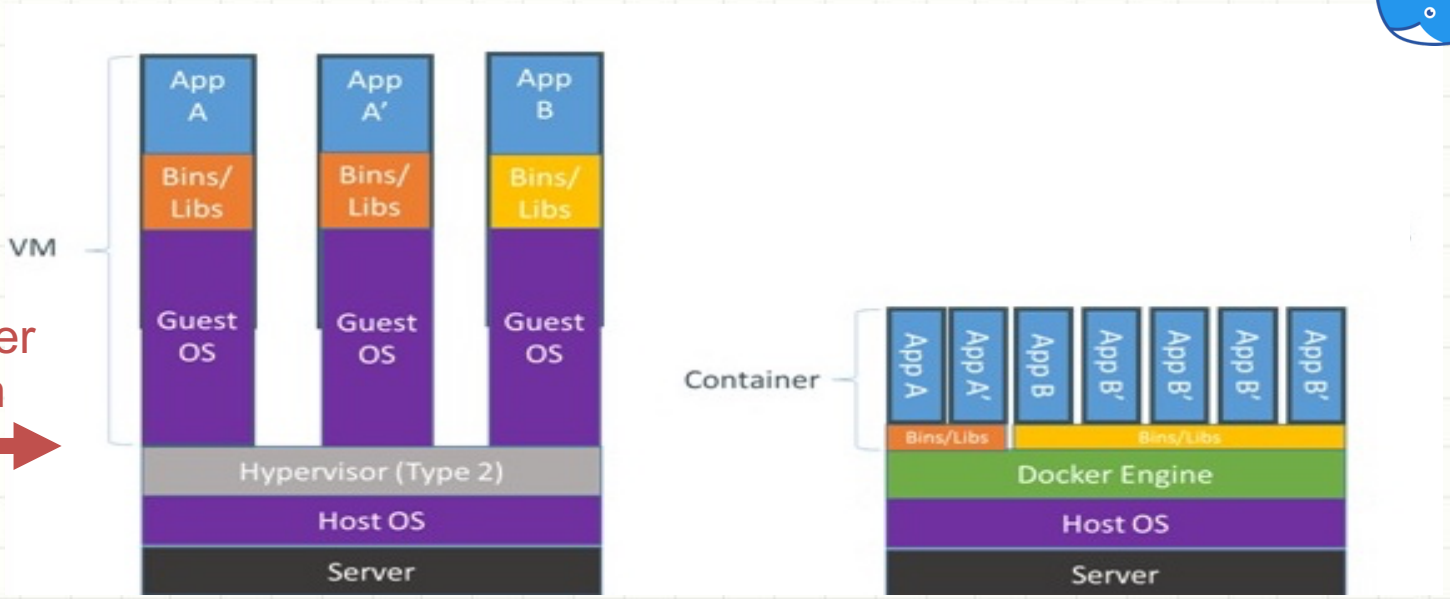
Features

Manages regionalized scaling of docker images

More complex

Flashback on docker

VMs vs. Containers

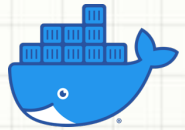


Docker key advantages:
- Better resources utilization (less overhead): CPU, RAM
- Faster to stop/start applications (seconds)
- Enable powerful portability

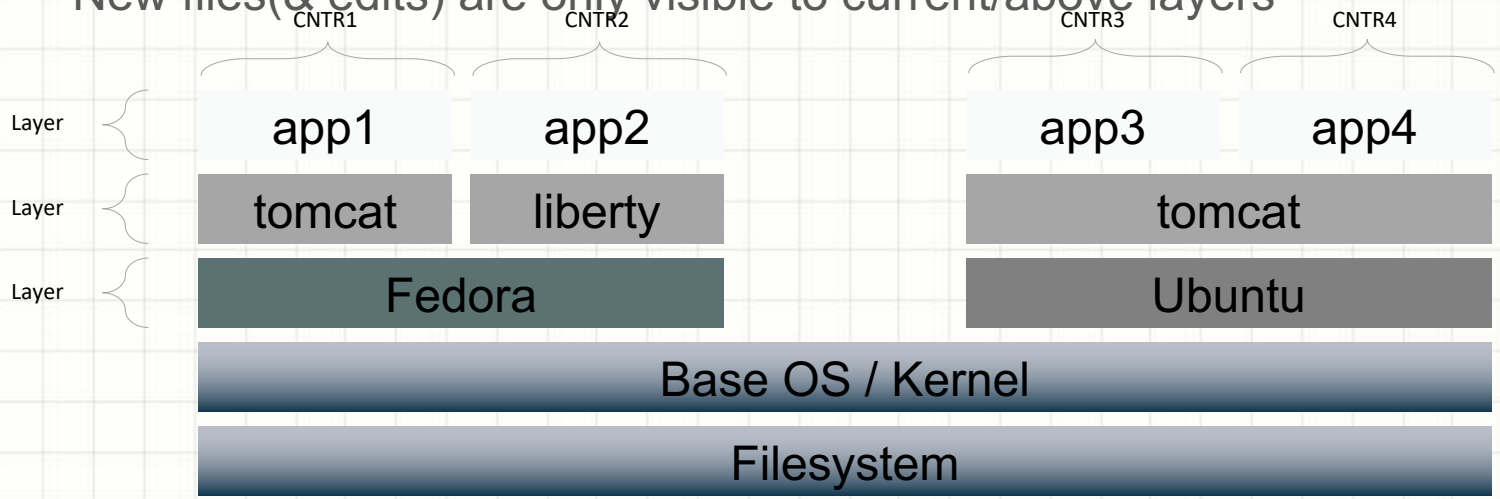
To be known :
- Fast change
- 'Dockerization' effort

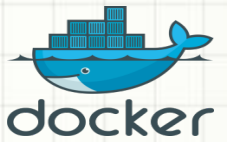
Docker Containers

A technical view into the shared and layered file systems technology



Docker uses a copy-on-write (union) filesystem
New files(& edits) are only visible to current/above layers





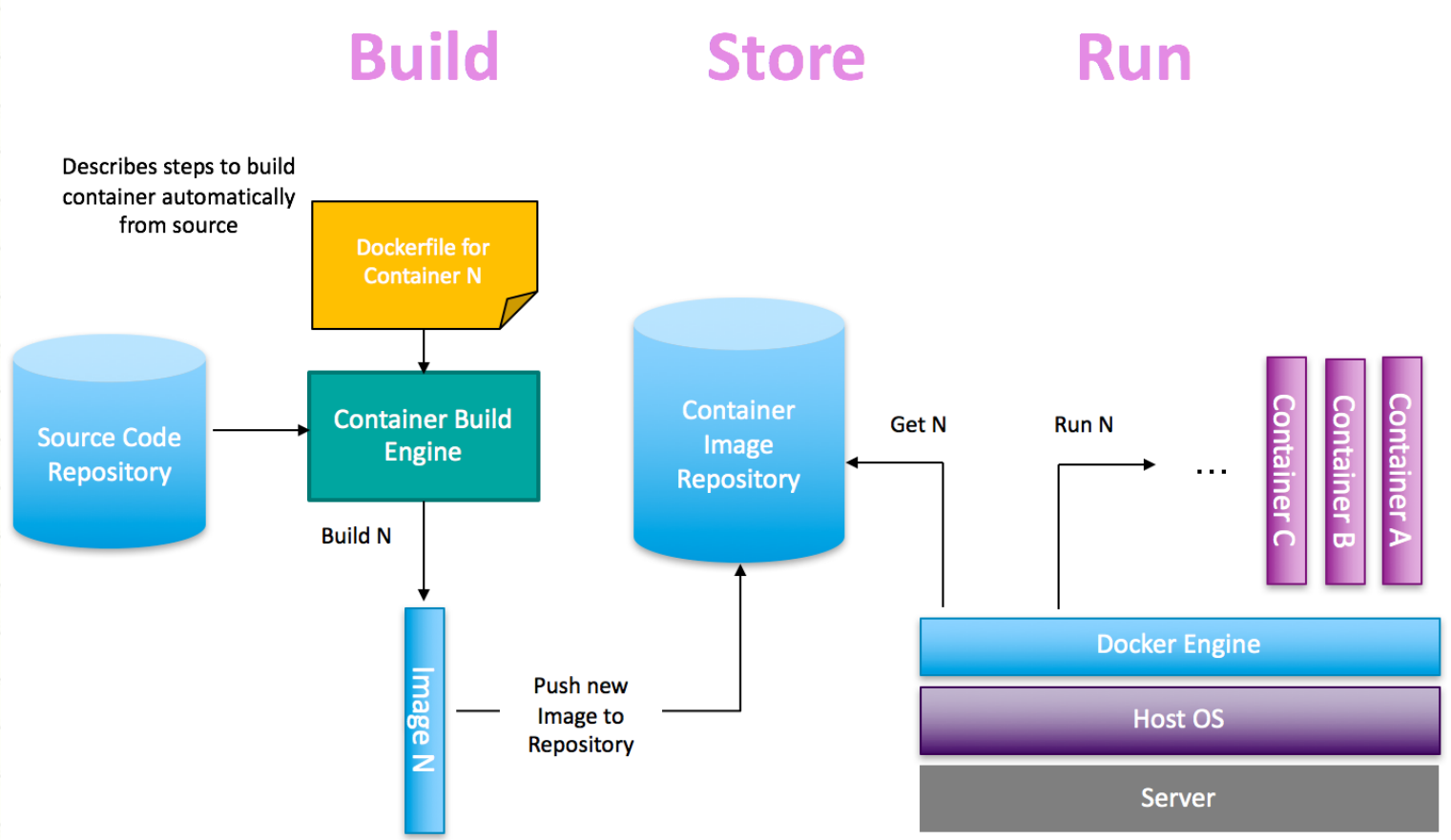
Layers allow for reuse

- More containers per host
- Faster start-up/download time – base layers are "cached"

Images

- Tarball of layers (each layer is a tarball)

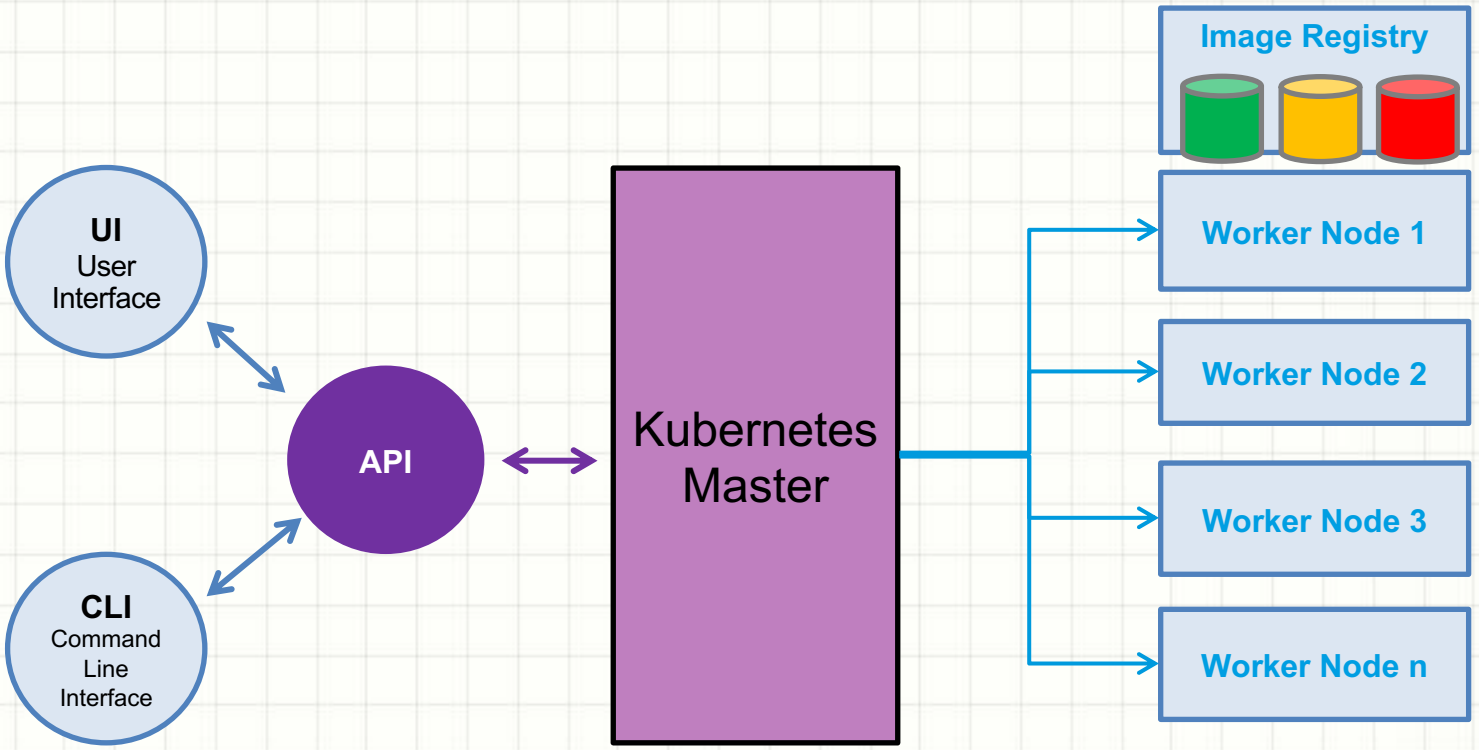
Docker basic functions



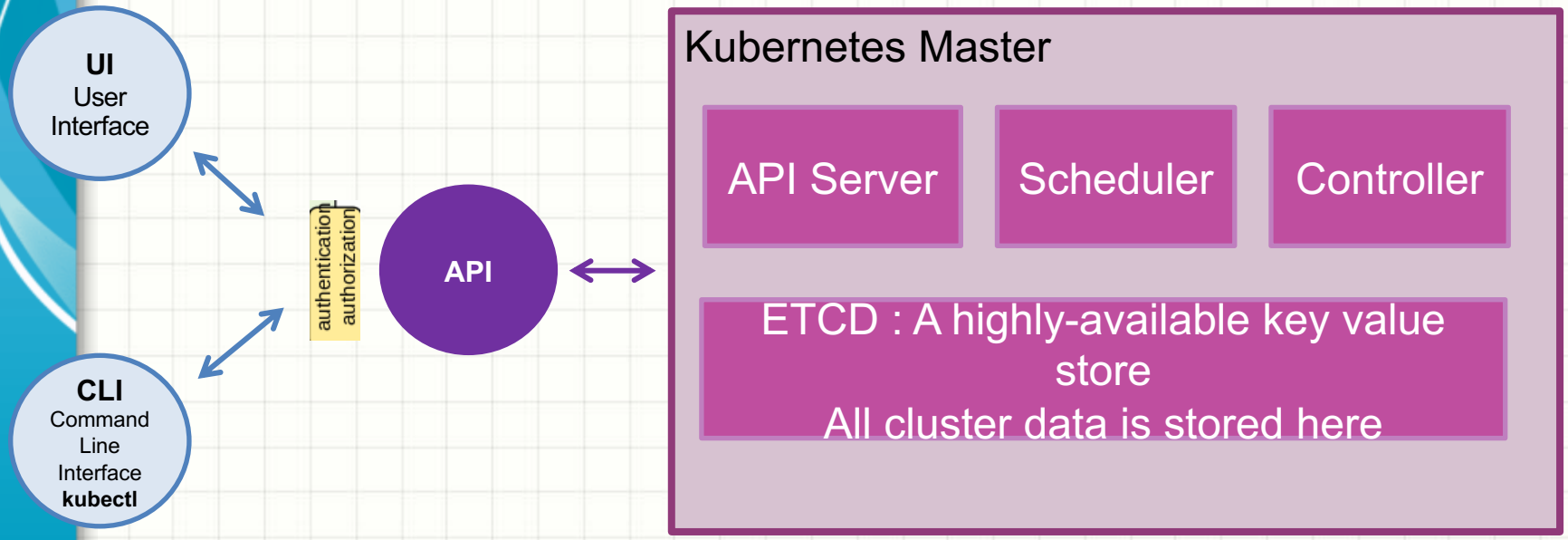
What is Kubernetes?

- Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure
  - Container orchestrator
  - Runs and manages containers
  - Supports multiple cloud and bare-metal environments
  - Inspired and informed by Google's experiences and internal systems
  - 100% Open source, written in Go
  - Manage applications, not machines
  - Rich ecosystem of plug-ins for scheduling, storage, networking

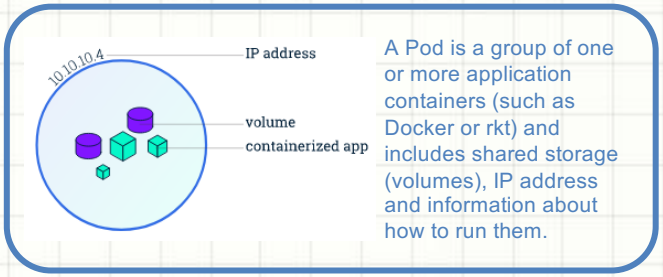
Kubernetes Architecture

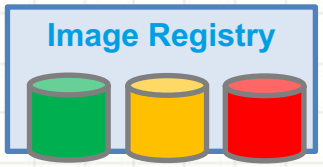


Kubernetes Architecture



Kubernetes Architecture





Kubernetes Master

Worker Node 1, 2, 3… n

POD = smallest deployment unit in K8S

Pod

Container

Pod

Container

Pod

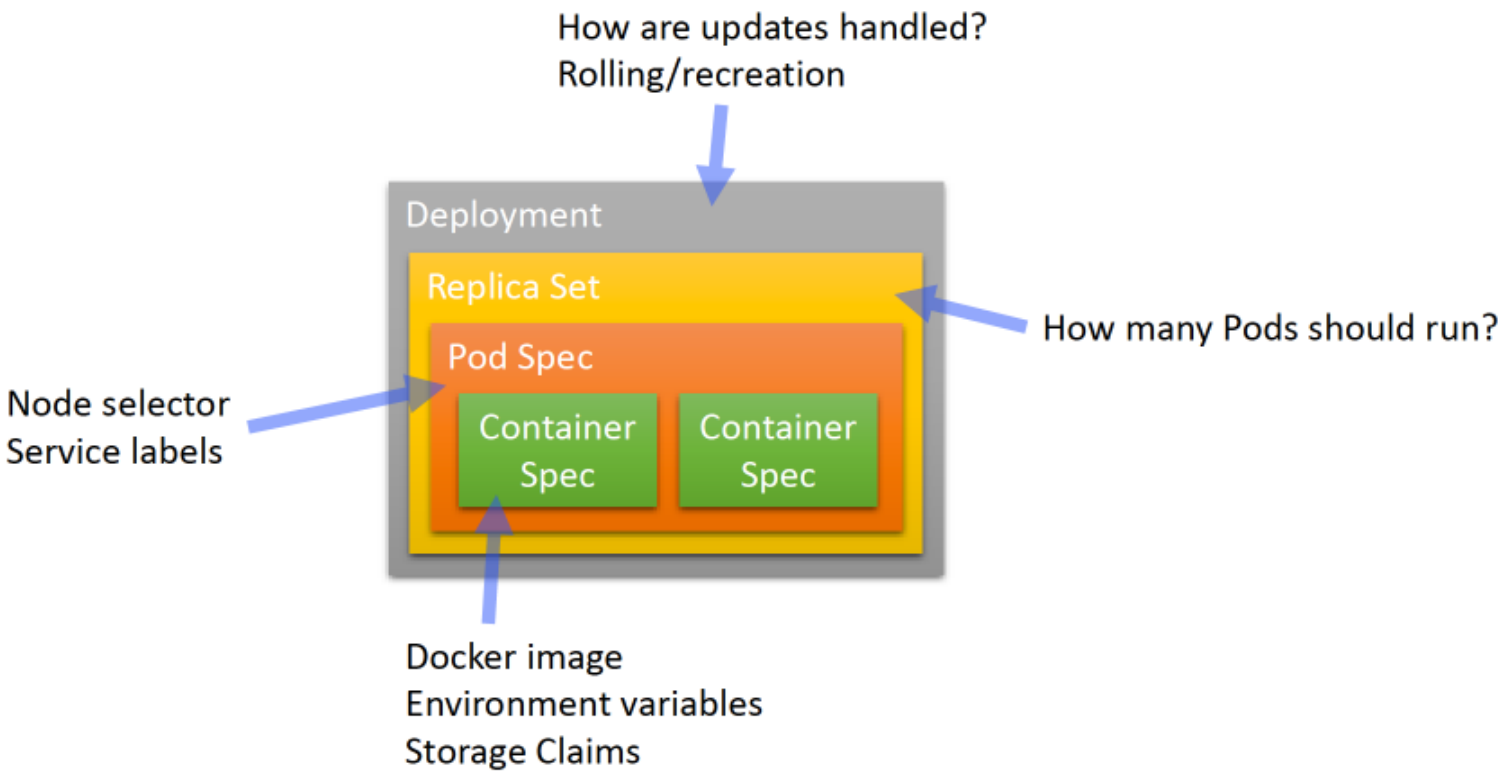
Container

docker

kubelet

Kube-proxy

Kubernetes Concepts: Deployment, ReplicaSet, Pod



Deployment and Service Yaml

```
apiVersion: apps/v1
 kind: Deployment
 metadata:
   name: mytodos
 spec:
   replicas: 2 # tells deployment to run 2 pods
   selector:
     matchLabels:
       app: mytodos
   template:   # create pods using pod
definition in this template
     metadata:
       labels:
         app: mytodos
         tier: frontend
     spec:
       containers:
       - name: mytodos
         image: <image>:1.0
         imagePullPolicy: Always
         resources:
           requests:
             cpu: 250m     # 250 millicores =
1/4 core
             memory: 128Mi # 128 MB
           limits:
             cpu: 500m
             memory: 384Mi
```

```
apiVersion: v1
 kind: Service
 metadata:
   name: mytodos
   labels:
     app: mytodos
     tier: frontend
 spec:
   ports:
   - protocol: TCP
     port: 8080
   selector:
     app: mytodos
     tier: frontend
```

Tutorial

- https://kubernetes.io/docs/tutorials/kubernetes-basics/

QUESTIONS?