



**kubernetes**

2018-02-16 @ Code Garden

**Baby steps standing on the shoulders of the giants**

# DAVID ROSSI

- sysadmin @ Unidata
- monitoringartist
- *devhopes*

# FRANCESCO ULIANA

## @WARRIOR10111

- technologist @ CNR
- java/scala
- devops

# AGENDA

- container orchestration
- getting started
- ecosystem / APIs
- kubernetes building blocks

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



# DOCKER JOURNEY



```
docker run nginx
```

moving to prod..

```
docker run --restart=always -d my-app  
# will break with docker upgrade  
  \_(ツ)_/
```

# systemd unit

## [Unit]

```
Description=Redis Container  
After=docker.service  
Requires=docker.service
```

## [Service]

```
TimeoutStartSec=0  
Restart=always  
ExecStartPre=-/usr/bin/docker stop %n  
ExecStartPre=-/usr/bin/docker rm %n  
ExecStartPre=/usr/bin/docker pull redis  
ExecStart=/usr/bin/docker run --rm --name %n redis
```

multiple services - how do I manage dependencies?

```
docker-compose up -d
```

- How to scale?
  - HA?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
  - health check?
- How to keep them running?
- How to update them?
  - rolling upgrades?
- Where are my containers?
- How do I manage secrets?

we need an orchestrator...but Kubernetes is SO HARD!



```
docker swarm init
```

```
docker swarm join --token SWMTKN-1-  
XYZ 192.168.X.Y:2377
```

```
docker service create --name app  
app:1.0
```

```
docker stack deploy -c app.yml app
```

- bad APIs
- ecosystem?
  - logging
  - monitoring
  - developer workflow
  - API gateway
- 3rd party software integration (CI/CD)?
- cloud provider support?
- difficult to troubleshoot

# ...in the end, Docker decided to support Kubernetes



## DOCKER PLATFORM AND MOBY PROJECT ADD KUBERNETES

By [Solomon Hykes](#)

October 17, 2017



containerd, Docker and Kubernetes, Docker CE, Docker EE, Kubernetes, Moby Project

Today we're announcing that the Docker platform is integrating support for Kubernetes so that Docker customers and developers have the option to use both Kubernetes and Swarm to orchestrate container workloads. [Register for beta access](#) and check out the detailed blog posts to learn how we're bringing Kubernetes to:

**FIRST STEPS**

# INSTALLING ON DEV MACHINE

- minikube
- kubectl

```
minikube start
```



```
minikube status
```

```
kubectl cluster-info
```

# BASIC COMMANDS

```
kubectl run webserver --  
image=nginx:alpine
```

```
kubectl port-forward nginx-  
8669f45f7d-4ldqc 8080:80
```

```
kubectl expose deployments webserver  
--port=80 --type=NodePort
```

```
kubectl get services
```

```
kubectl logs -f webserver-XXX
```

```
kubectl exec -it webserver-XXX -- sh
```



```
kubectl scale deployment --replicas=4  
webserver
```

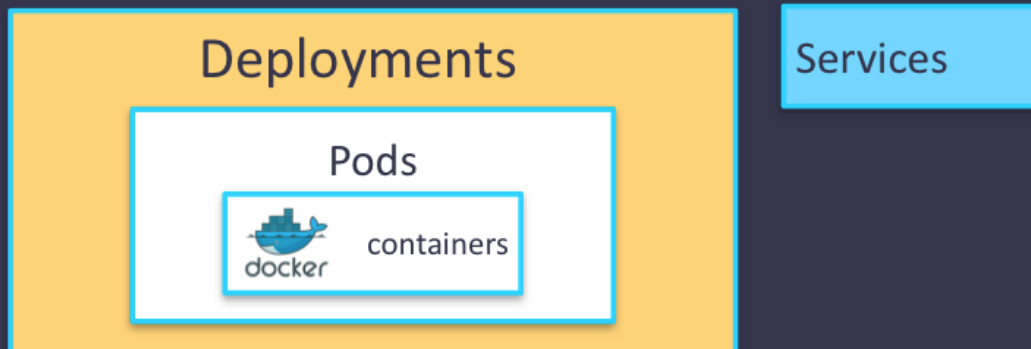
minikube dashboard

minikube service monitoring-grafana -  
n kube-system

```
kubectl set image  
deployment/webserver  
webserver=nginx:1.10-alpine --  
record=true
```

```
kubectl rollout history deployment  
webserver
```

# Recap: all you need to know



<b>Container Image</b>	Docker container image, contains your application code in an isolated environment.
<b>Pod</b>	A set of containers, sharing network namespace and local volumes, co-scheduled on one machine. Mortal. Has pod IP. Has labels.
<b>Deployment</b>	Specify how many replicas of a pod should run in a cluster. Then ensures that many are running across the cluster. Has labels.
<b>Service</b>	Names things in DNS. Gets virtual IP. Two types: ClusterIP for internal services, NodePort for publishing to outside. Routes based on labels.

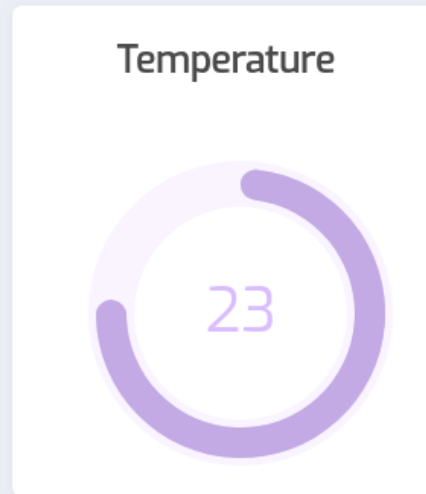
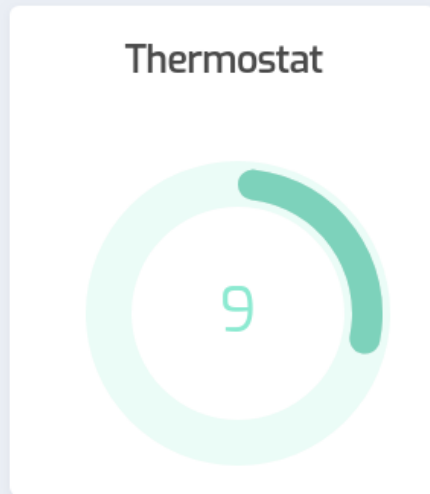
## pods vs containers

- sidecar containers (e.g. agent), proxies, etc.

# CASE STUDY



- `kubectl get deployment webserver -- output=yaml`
- ide support



<https://github.com/francescou/kubernetes-code-garden>

**PRODUCTION**

horizontalpodautoscaler

secrets

namespaces

network policies

## Operators

"An Operator represents human operational knowledge in software to reliably manage an application."



# ECOSYSTEM

- openshift
- efk
- helm
- istio
- draft
- spinnaker
- function-as-a-service

**STATEFUL CONTAINERS?**



**Kelsey Hightower** ✓

@kelseyhightower

Following



Kubernetes has made huge improvements in the ability to run stateful workloads including databases and message queues, but I still prefer not to run them on Kubernetes.

🌐 Traduci dalla lingua originale: inglese

15:04 - 13 feb 2018

222 Retweet 509 Mi piace

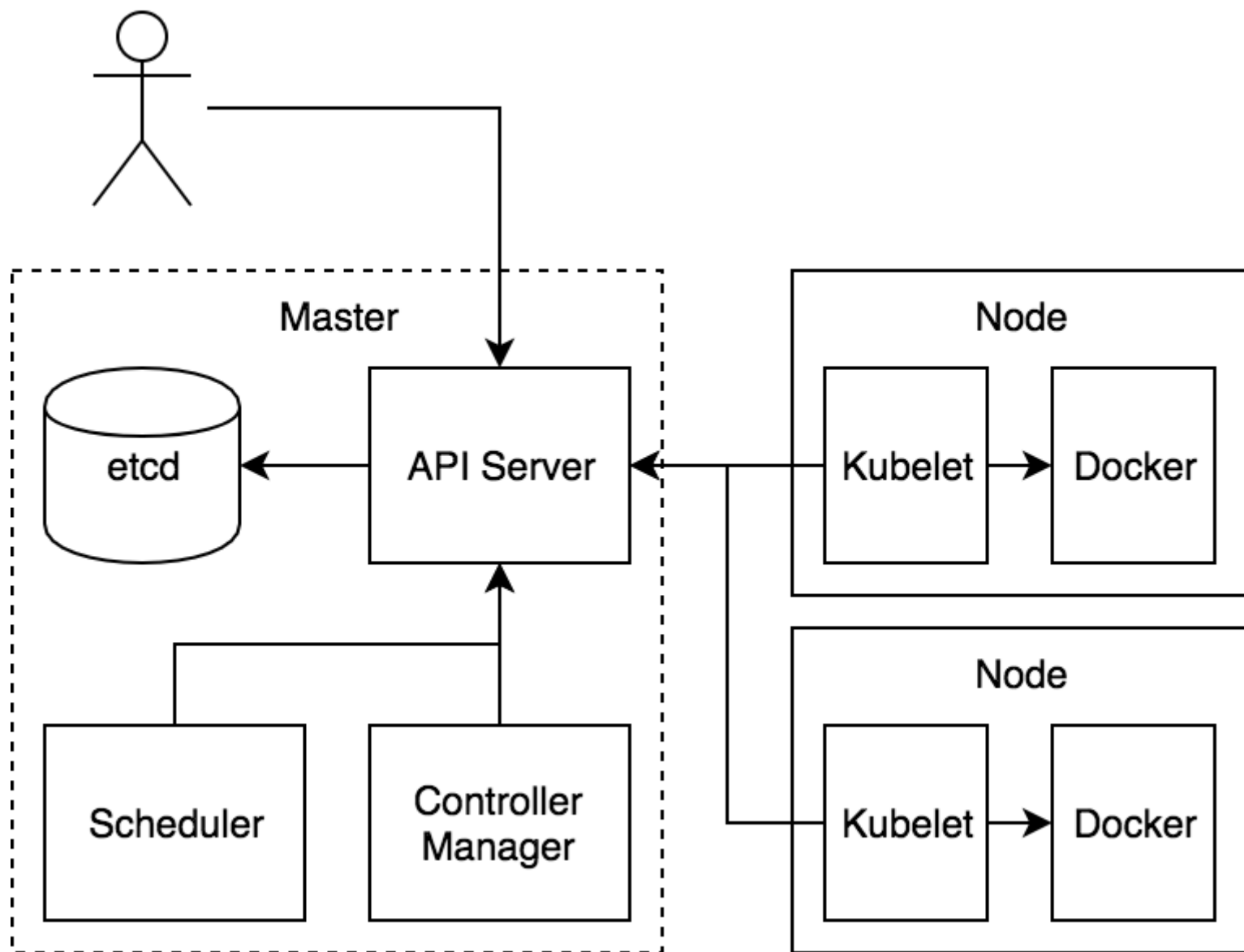


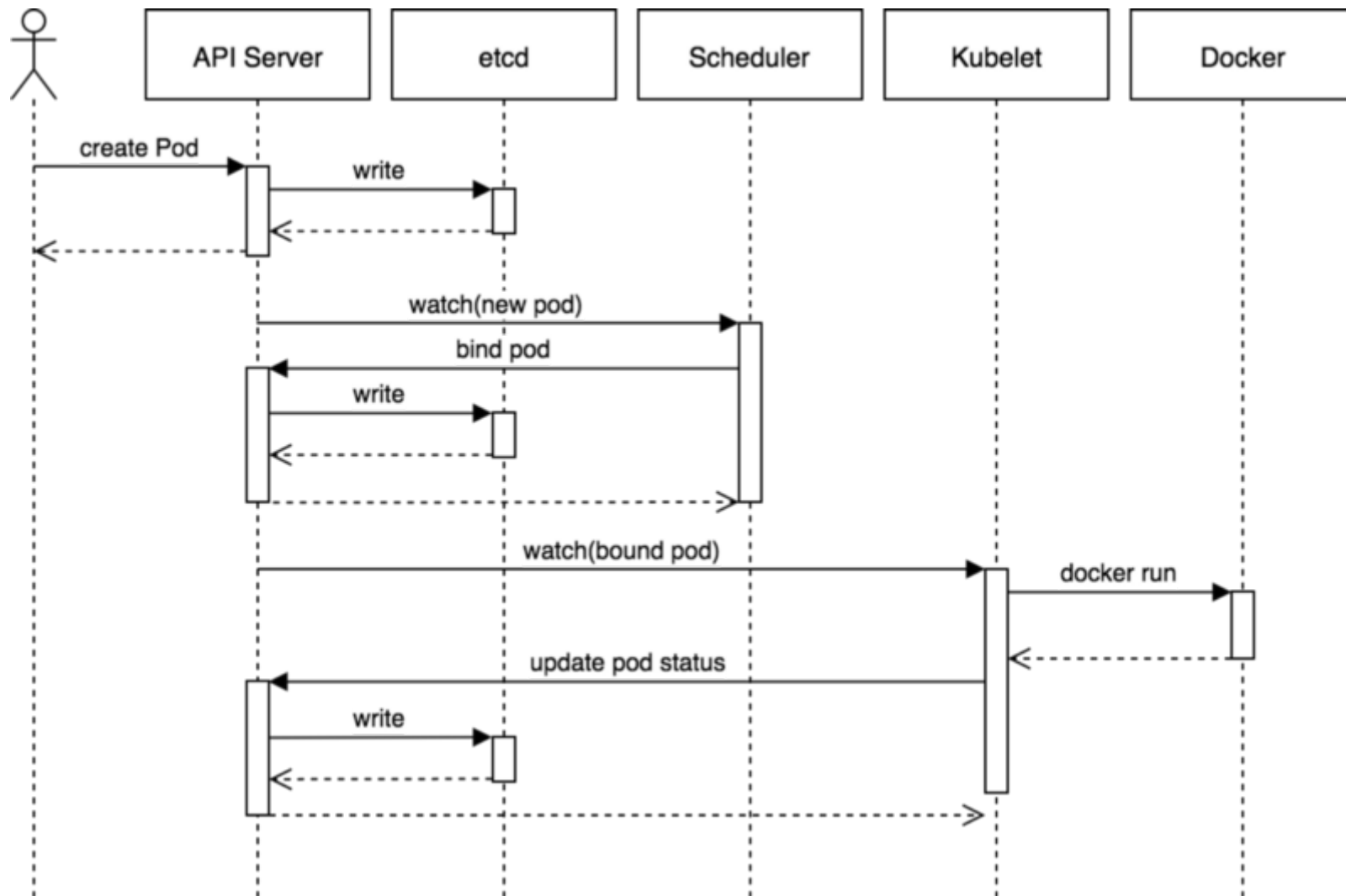
💬 21 ↺ 222 ❤️ 509 ✉

**UNDER THE HOOD**

# KUBERNETES KEY ELEMENTS

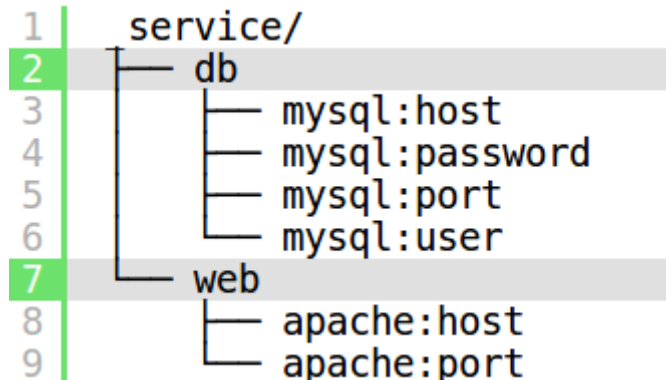
- etcd
- kubelets
- container runtime
- CNI (SDN)





# ETCD KEY STORE FEATURES

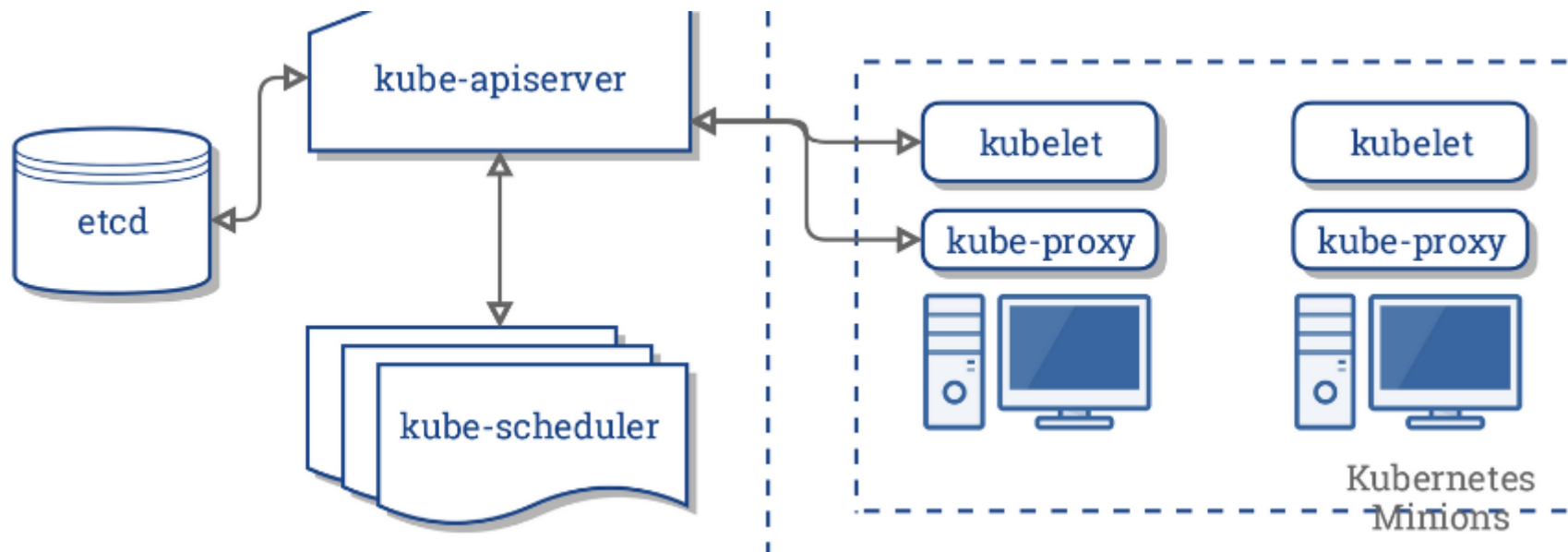
- dynamic
- distributed
- group fs like values
- TTL values
- order recording





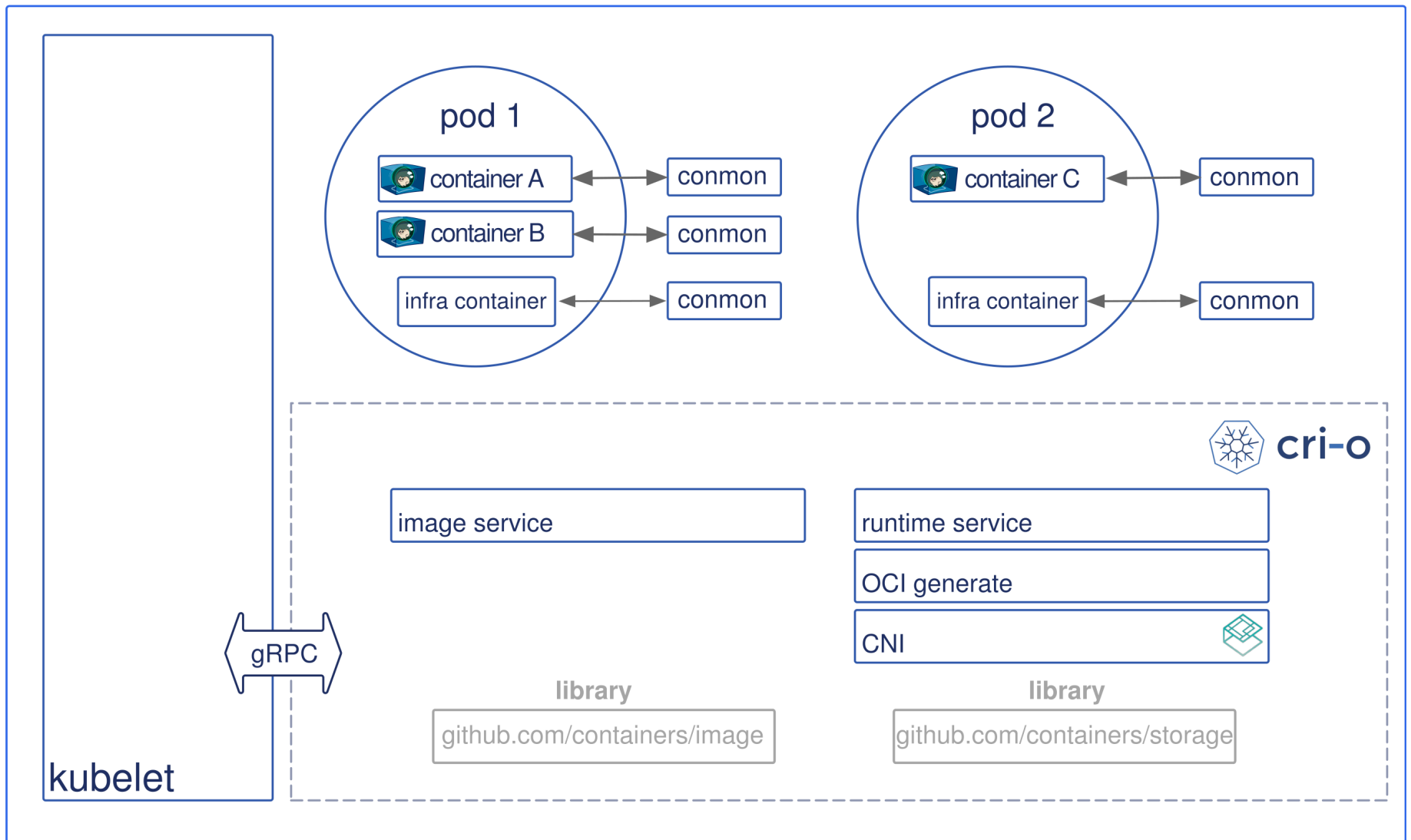
# KUBELET: NODE AGENT THAT STARTS PODS AND CONTAINERS

- lowest level component (I will touch)
- responsible for what's running on an individual machine
- unit of execution: pod
- consumes other plugins
- kubelet introspection (:10255/healthz)



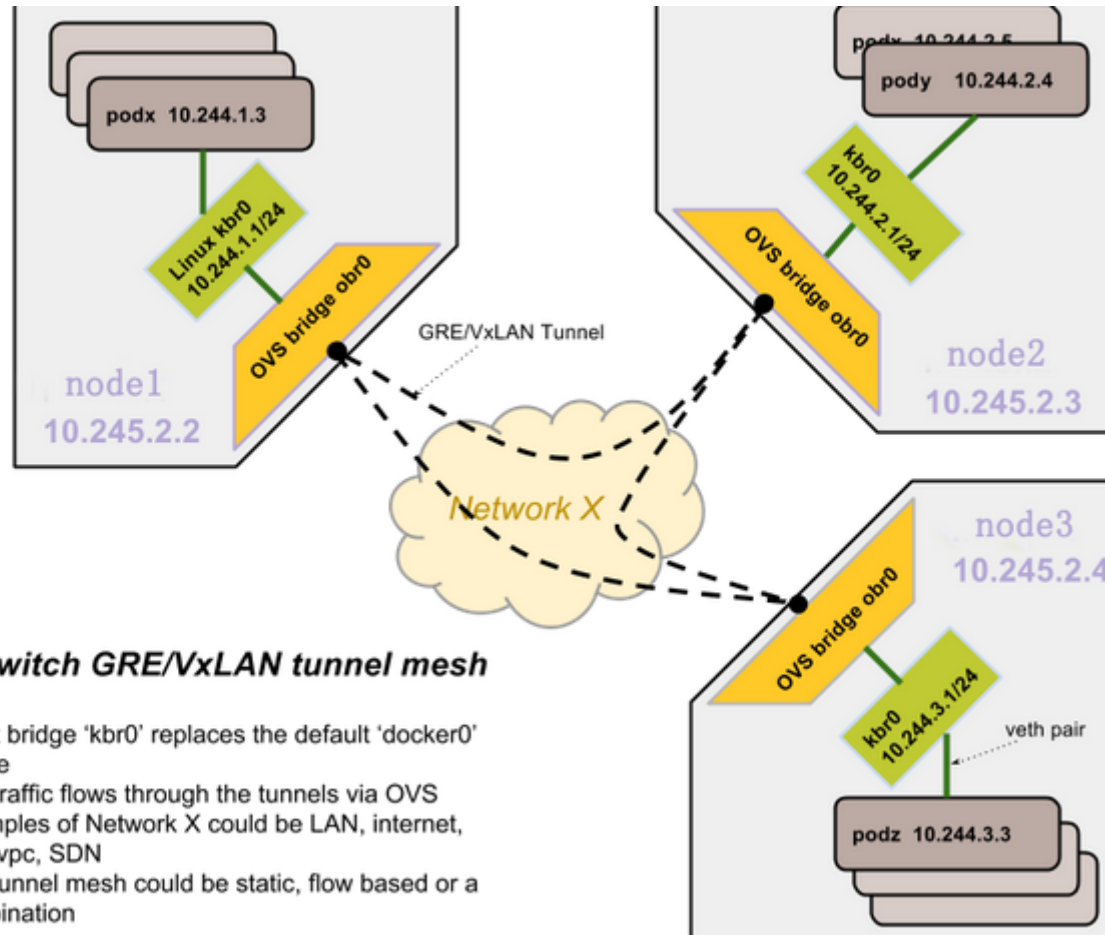
# OPEN CONTAINER INITIATIVE

- plugin runtime interface
- multiple runtimes OCI-compliant
- run containers directly from Kubernetes



# CONTAINER NETWORK INTERFACE & SDN PLUGIN

- Hooked by container runtime
- CNI: API to call networks plugins
- network plugins
  - kubenet (default linux bridge)
  - flannel (l3 SDN)
  - overlay networks
    - ovswitch (l2 SDN)



### OpenVSwitch GRE/VxLAN tunnel mesh

- Linux bridge 'kbr0' replaces the default 'docker0' bridge
- Pod traffic flows through the tunnels via OVS
- Examples of Network X could be LAN, internet, EC2 vpc, SDN
- The tunnel mesh could be static, flow based or a combination

# FLANNEL SDN PLUGIN

- arm friendly (rpi ref)
- simple deploy:

```
kubectl apply -f \  
  https://raw.githubusercontent.com/coreos/flannel/master/Do
```

```
kubectl apply -f \  
  https://raw.githubusercontent.com/coreos/flannel/master/Do
```



## kubernetes components

JULIA EVANS  
@b0rk


★ means "required to run pods + daemonsets" -- set these up first!

● : required for some but not all API resources.

### master components

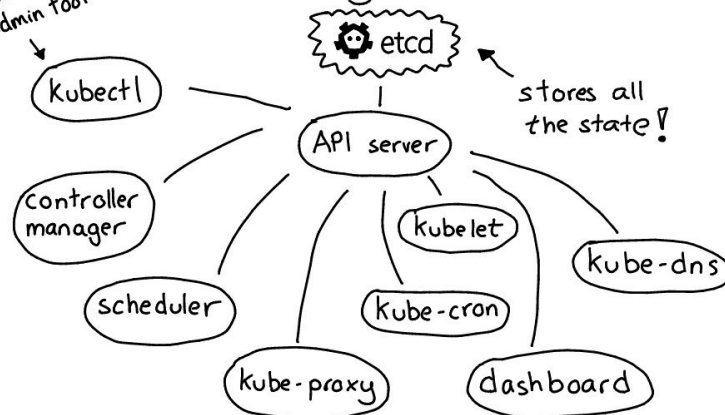
- ★  etcd
- ★ API server
- ★ controller manager
- ★ scheduler
- kube-cron  
(alpha, for cron jobs)
- kube-dns  
(for Service DNS names)
- dashboard  
(a web interface!)

### node components (run on every node)

- ★  docker daemon
- ★ kubelet  
(Docker + kubelet run ~ all other components)
- supervisord / systemd  
(to keep kubelet + docker running)
- kube-proxy  
(makes Service IPs work)

command line  
admin tool

### How they fit together





# CONCLUSIONS

- when to use k8s
  - world scale :)
  - ecosystem
  - reliability
  - fully managed kubernetes
    - AWS EKS
    - AKS
    - GKE

- when not to use k8s
  - operational complexity, lots of moving parts
  - you'll need a dedicated kubernetes team
  - KISS



**Kelsey Hightower** ✓

@kelseyhightower

Following



Avoid when:

- You require a non Linux or Windows OS
- A fully managed PaaS or FaaS works
- Your workload was designed to run on a Mainframe
- You only need to host a personal WordPress blog
- What you have already works

**Iván Mosquera** @ivmos

In risposta a @kelseyhightower

So when should I avoid k8s? :)

🌐 Traduci dalla lingua originale: inglese

17:37 - 13 feb 2018

225 Retweet 687 Mi piace



22



225



687



# RESOURCES

- [https://jvns.ca/blog/2017/10/05/reasons-kubernetes-](https://jvns.ca/blog/2017/10/05/reasons-kubernetes-is-awesome/)
- [https://www.nginx.com/resources/library/kubernetes developers/](https://www.nginx.com/resources/library/kubernetes-developers/)
- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- <https://docs.google.com/presentation/d/1ij64THksTygN0ipc6MDF4cGBRQcV3BRYaoM/edit#slide=id.g25913a8d0e444f0b0000000000000000>
- <https://github.com/jamiehannaford/what-happens-when-you-run-k8s/blob/master/README.md>

- <https://github.com/kubernetes/community/blob/master/proposals/architecture/architecture.md>
- <https://chandanduttachowdhury.wordpress.com/tag/>
- <http://kamalmarhubi.com/blog/2015/08/27/what-ever>
- <https://www.redhat.com/en/blog/running-production-containers-introducing-ocid>
- <https://github.com/containernetworking/cni>

- <https://www.slideshare.net/inwinstack/network-plug-kubernetes>
- <http://nerdrug.org/blog/introduzione-a-open-vswitch>
- <https://blog.hypriot.com/post/setup-kubernetes-rasp-cluster/>
- <https://twitter.com/b0rk/status/872822361199972352>



**QUESTIONS ?**