



MIRANTIS

Agnostic intro to K8s without BS
PART 1

Fabrizio Soppelsa

16.03.2017

Part 1 AGENDA

- K8s fundamental concepts
- Playing with containers on K8s
- Hands-on demo
- Basic K8s concepts
- Q&A

Kubernetes fundamentals

An open source *container cluster* manager

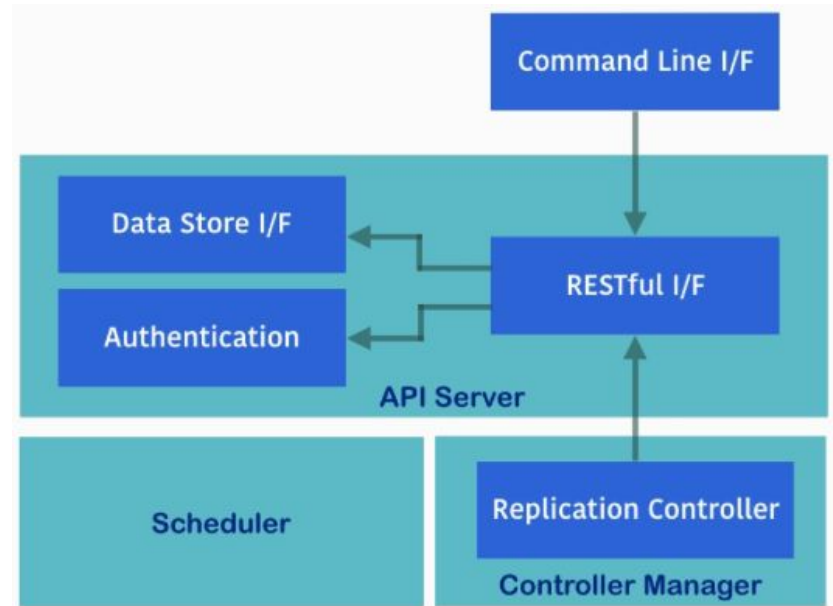
Part of the Cloud Native Computing Foundation initiative

Kubernetes structure

- Kubernetes **master**
- Kubernetes **nodes**
- **etcd**
- Some SDN
- Linux as a base host
- Typically uses Docker (but...)

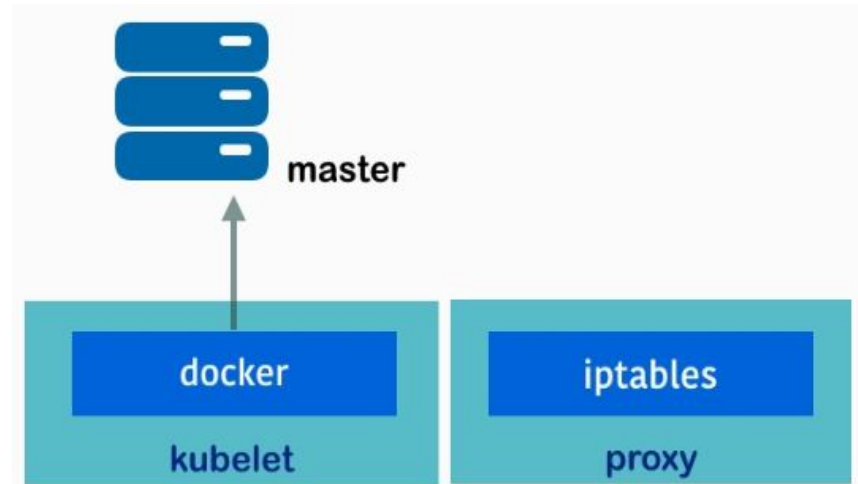
Kubernetes master node

- Client - **kubectl**
- Running services
 - **kube-apiserver**
 - **kube-scheduler**
 - **kube-controller-manager**



Kubernetes node

- Running services
 - **kubelet**
 - **kube-proxy**



- Distributed key-value store
- *External* to K8s (dependency)
- `apt-get install etcd, dnf install etcd`
- `etcdctl` interface
- `/etc/etcd/etcd.conf` sample (for non-HA setups):

```
ETCD_NAME=default  
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"  
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"  
ETCD_ADVERTISE_CLIENT_URLS="http://0.0.0.0:2379"
```

Kubernetes concepts

- **Control** is for nodes operations
- **Pods** are the smallest deployable unit
- A **replication controller** governs pods replicas and scale
- A **service** may describe a multi-tier app
- **Volumes** for storing persistent data (stateful)
- **Secrets, namespaces, labels**, and more concepts.
- YAML everywhere

Some hands-on

Playing with pods and containers
A practical overview of K8s features

001 how to create a pod

```
apiVersion: v1
kind: Pod
metadata:
  name: my-first-pod
spec:
  containers:
    - name: my-nginx
      image: nginx
    - name: my-centos
      image: centos
      command: ["/bin/sh", "-c", "while : ;do curl http://
localhost:80/; sleep 3; done"]
```

```
# kubectl create -f 001-my-first-pod.yaml
```

002 replication controllers

```
kubectl run my-first-rc --image=nginx
```

```
kubectl run nginx-rc --image=nginx --replicas=2 --port=80
```

```
kubectl edit deployment/nginx-rc
```

```
kubectl delete deployment nginx-rc
```

003 services

```
kubectl run nginx-pod --image=nginx --port=80 --restart="Never" --labels="app=nginx"  
kubectl expose pod nginx-pod --port=8000 --target-port=80 --name="service-pod"
```

004 volumes

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
spec:
  containers:
  - image: ubuntu
    command:
    - sleep
    - "3600"
    imagePullPolicy: IfNotPresent
    name: ubuntu
  volumeMounts:
  - mountPath: /data-mount
    name: data
  volumes:
  - name: data
    emptyDir: {}
```

```
# kubectl create -f 004-volumes.yaml
```

005 rolling updating a rc

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-rc
spec:
  replicas: 2
  selector:
    role: load-balancer
  template:
    metadata:
      labels:
        role: load-balancer
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

```
kubectl rolling-update nginx-rc --image=fsoppelsa/nginx1
```

Thank you Q&A

Next week part 2
How to form a K8s cluster
And HA concepts