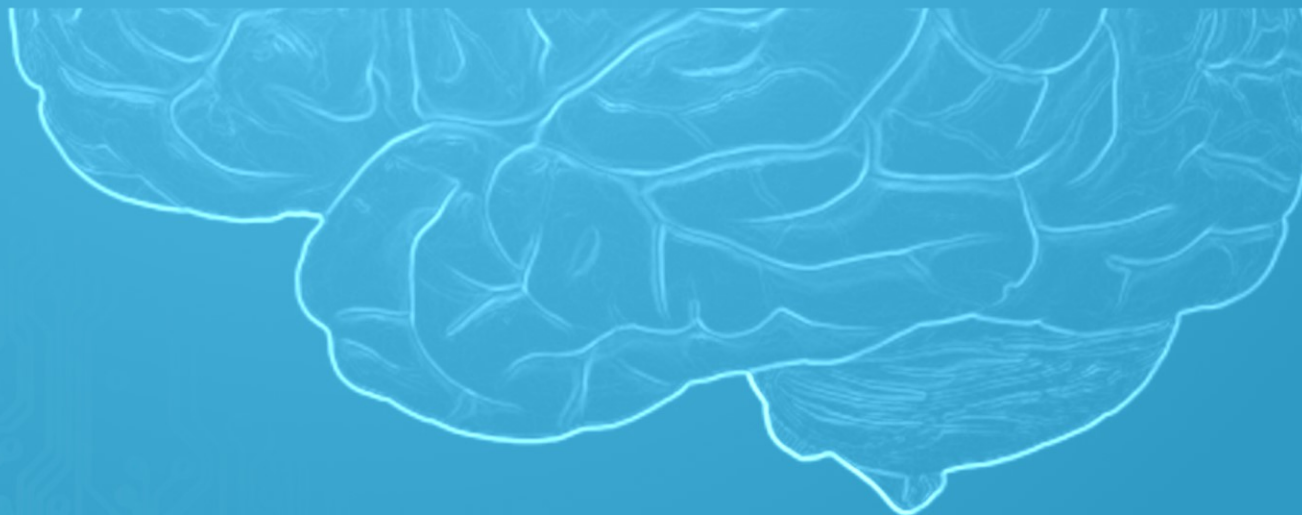




# Sisteme Tolerante la Defecte Orchestrare de Containere - Kubernetes

Lect. Dr. Ing. Cristian Chilipirea – [cristian.chilipirea@mta.ro](mailto:cristian.chilipirea@mta.ro)







# Containere



20 Martie 2013



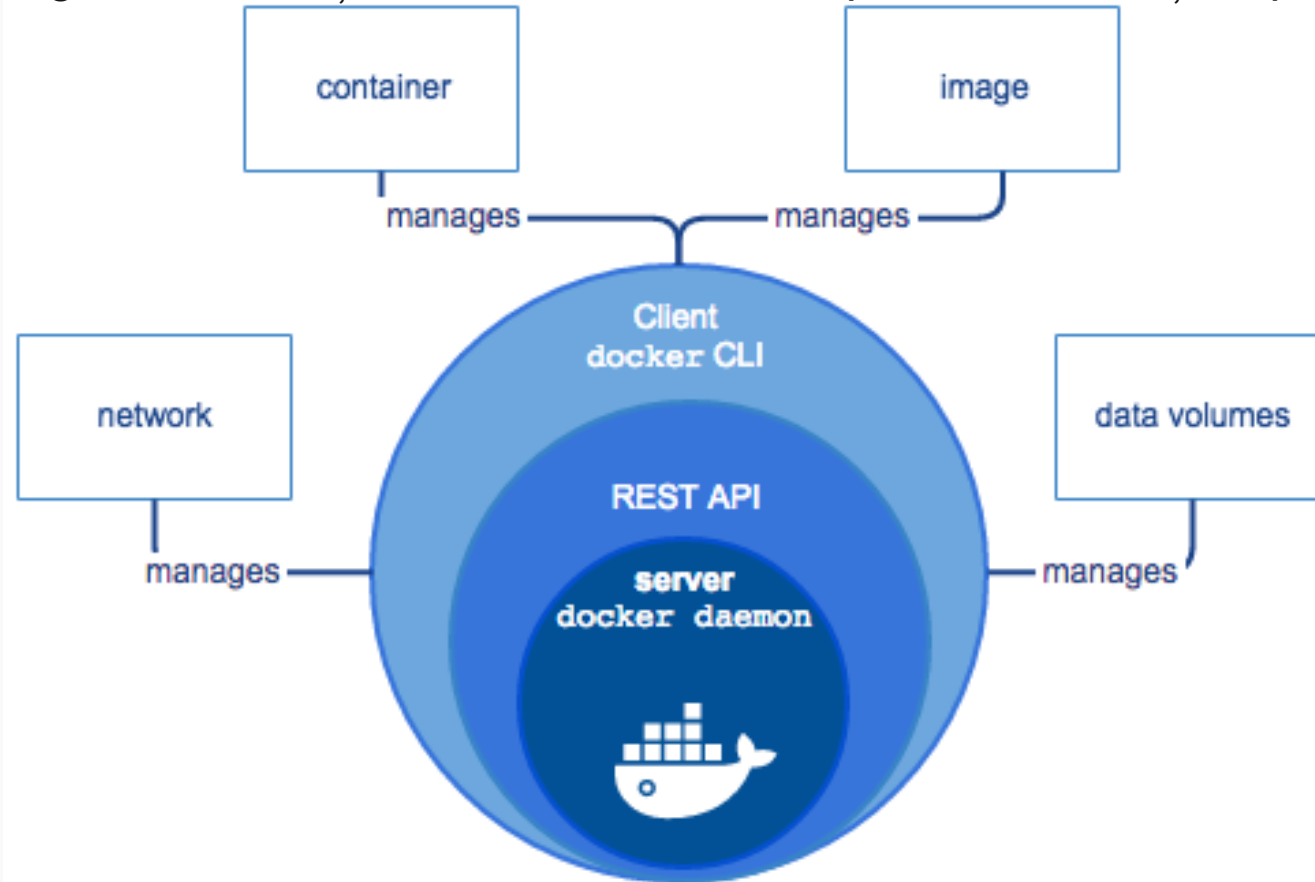
**Solomon Hykes**



# Docker Engine

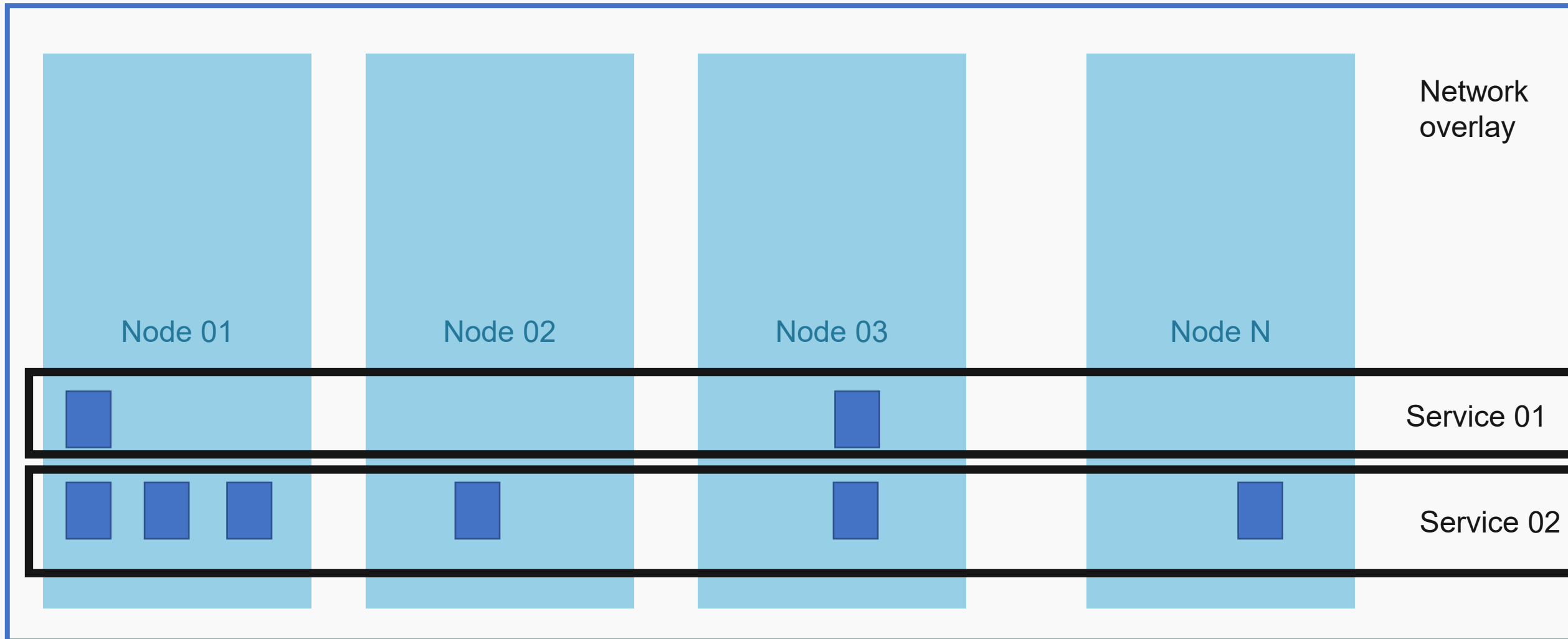
O imagine în execuție

Template cu instrucțiuni pentru rulare





# Docker Swarm





# Orchestrare containere - Kubernetes



10 July 2015

Google



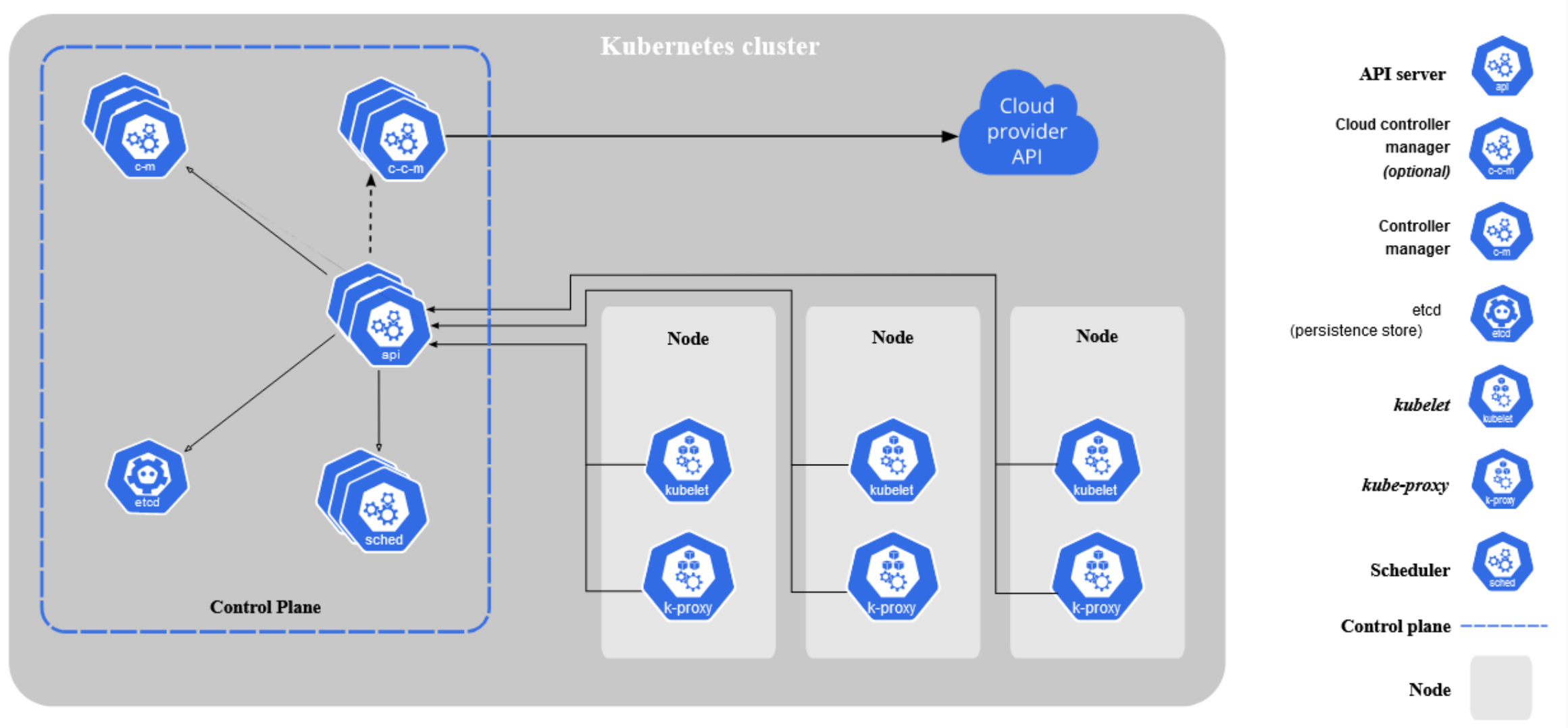
# Vocabulary

- **Nod**: mașină pe care rulează containere
- **Pod**: unul sau mai multe containere (containerele pot fi rulate doar în pod)
- **Controller**: deployment, replicaset, stateful set, daemon set, job, cronjob, etc.
- **Service**: network endpoint
- **Namespace**: filtru --- foarte diferit de namespace de la docker
- **Secrets**: parole, key ssh, necesare în containere, controlate de Kubernetes





# Kubernetes cluster







# Ce face Kubernetes?

- **Service Discovery** și **load balancing**
- **Storage orchestration**
- **Automated rollout** și **rollbacks**
- **Automatic bin packing** – în funcție de resursele puse la dispoziție pune containerele pe noduri în mod eficient
- **Self-healing** – dacă un container se oprește îl pornește automat
- **Secret** și **configuration management**



# Workloads

- **Deployment > ReplicaSet** – containererele pot fi ușor schimbate cu alte containere identice
- **StatefulSet** - containererele au atașat storage care trebuie reconectat la container dacă este oprit/repornit
- **DaemonSet** - utilitare necesare fiecărui nod
- **Job și CronJob** - rulează până termină munca apoi se opresc



# Useful commands

- `kubectl get all`
- `kubectl describe`
- `kubectl apply`
- `kubectl create deployment NAME --image IMAGE-NAME`
- `kubectl delete`
- `kubectl expose`
- `kubectl scale`
  
- `kubectl apply -R -f FOLDER/`
- `kubectl diff -R -f FOLDER/`



# Imperativ vs Declarativ

- Dai comenzi
- Descrii cum ar trebui să arate clusterul într-un fișier



# .yaml files

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```





# Play With Kubernetes

- <https://labs.play-with-k8s.com/>
- Necesită un cont Docker (free)
- Se poate să necesite să opriți add-blocker
  
- <https://www.katacoda.com>