



Kubernetes

Yassine Akharaze
Mohamad Mroue

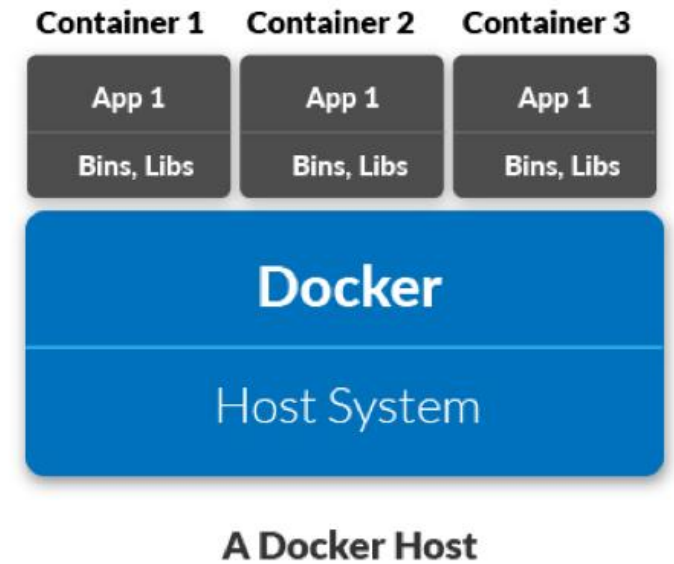
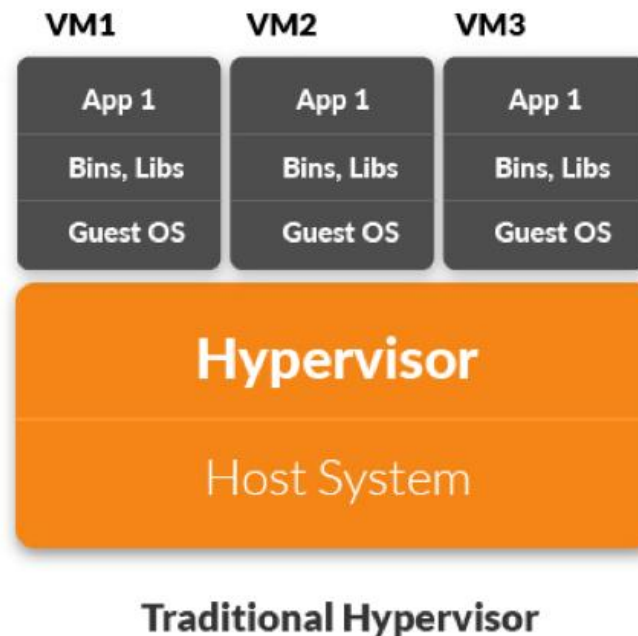
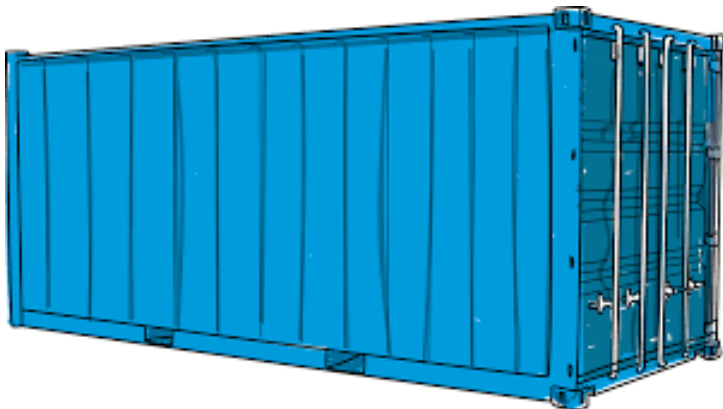
Plan

- Introduction
- Problématique
- Orchestration de conteneurs
- Kubernetes
- Architecture
- Démo
- Conclusion

Introduction

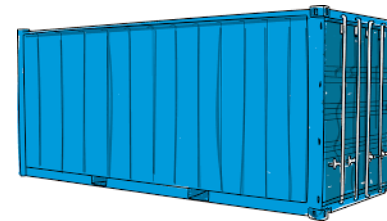
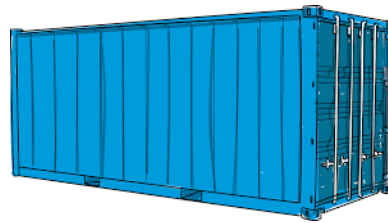
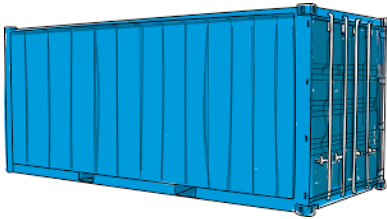
How do you package and distribute an application?

- Rapidité et facilité de déploiement des applications
- Portabilité
- Orienté micro-services



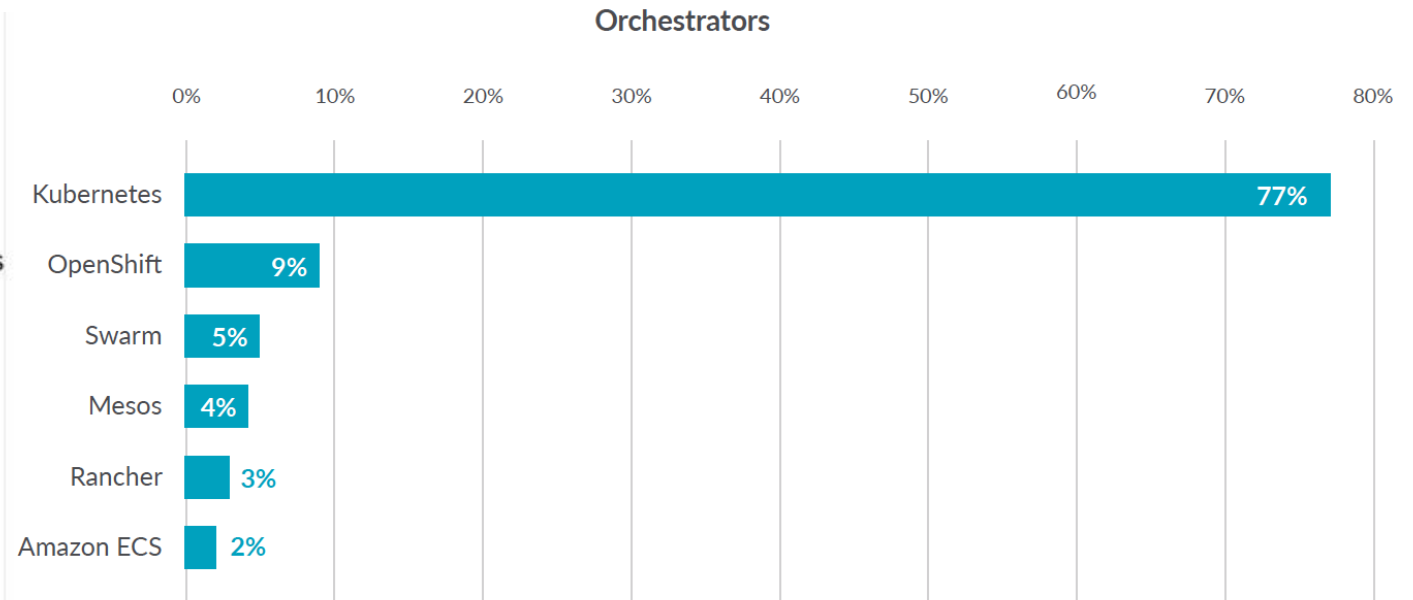
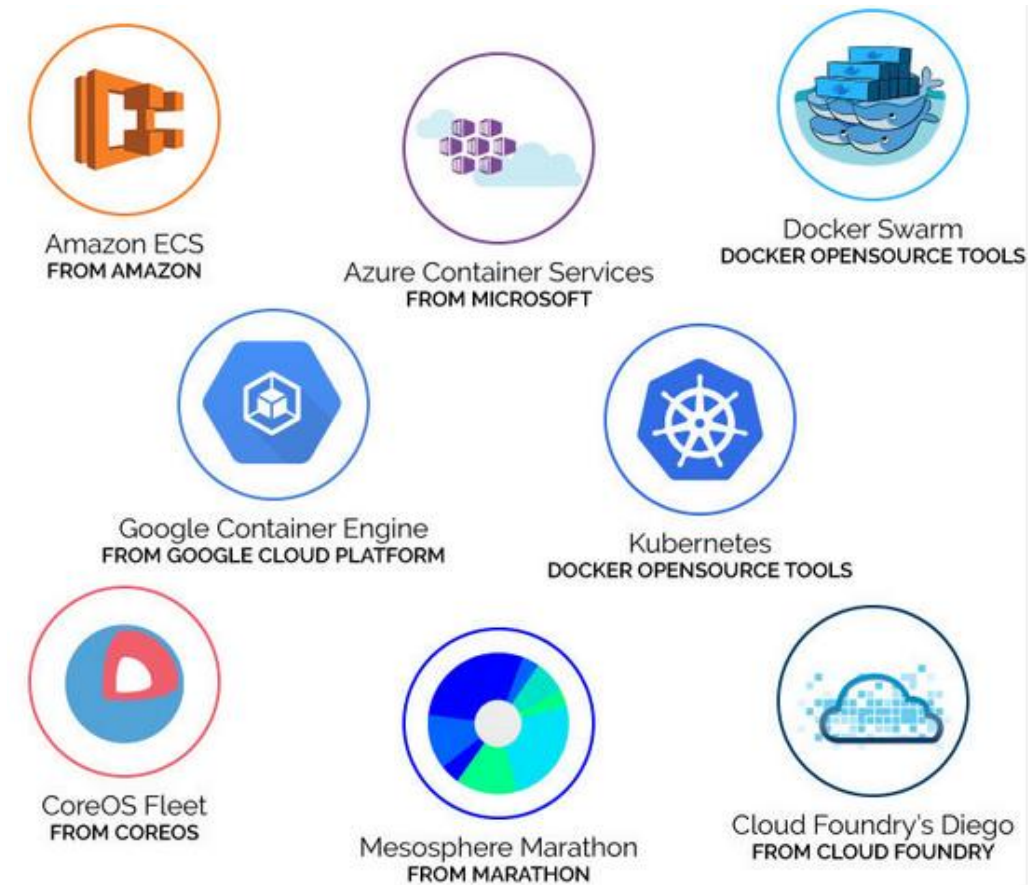
Problématique

- Monter en charge
- Gérer et monitorer



How do you scale, run and monitor an application?

Orchestrateur de conteneurs



Kubernetes



- **Orchestrateur** de conteneurs
- Open-source
- Créé par Google basé sur Borg
- Offert à la Cloud Native Computing Foundation
- Peut tourner en local, sur le cloud, Machines physiques
- **Gérer directement des applications et non plus des machines**
- **Principe : déclarer la cible à atteindre et laisser le système la maintenir**

Kubernetes: pourquoi ?

How do you scale, run and monitor an application?



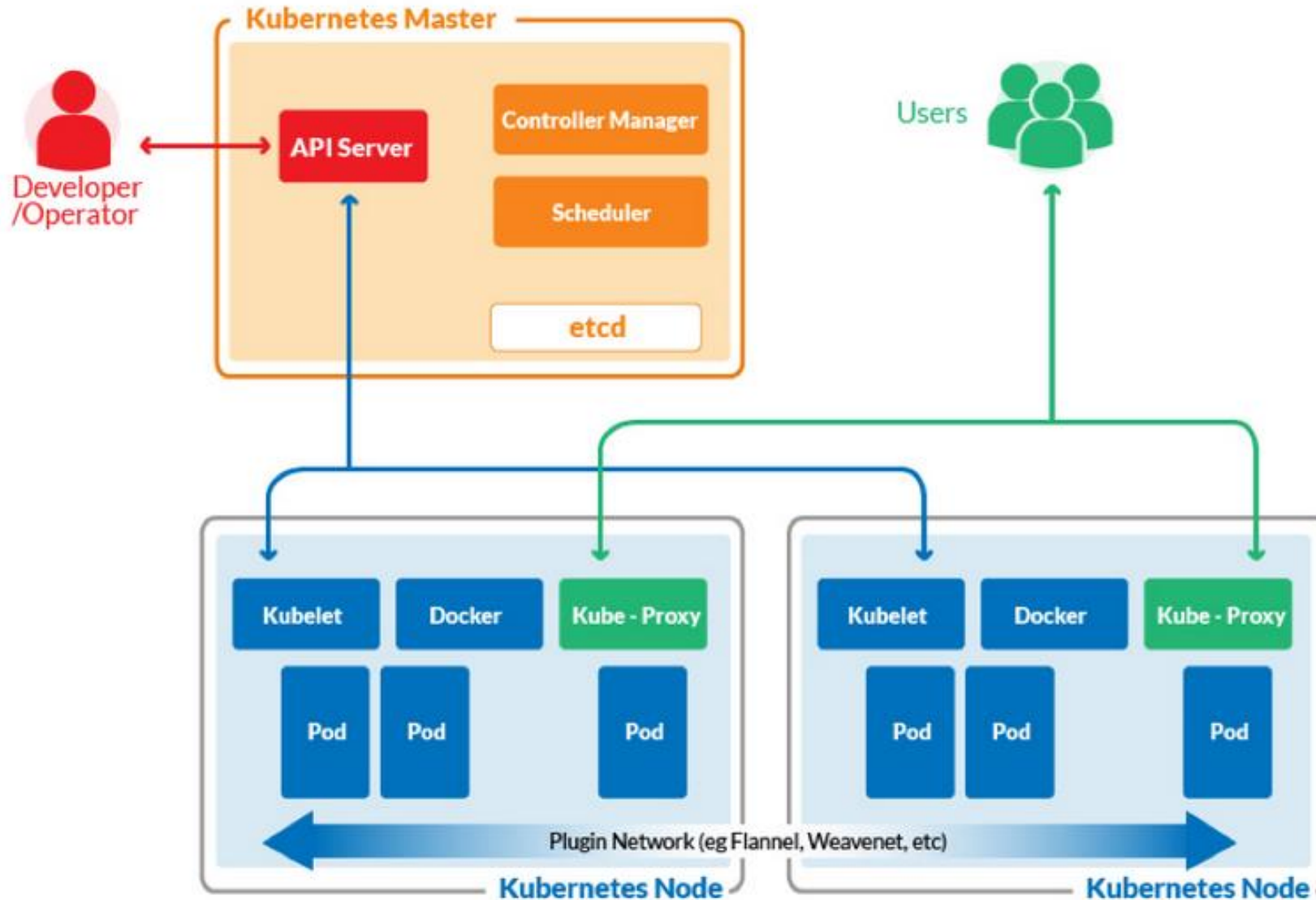
Que fait Kubernetes ? (1)

- **Déployer** une application
- **Automatiser** le déploiement et la Réplication de containers
- Organiser les containers en groupes et faire du **Load Balancing**

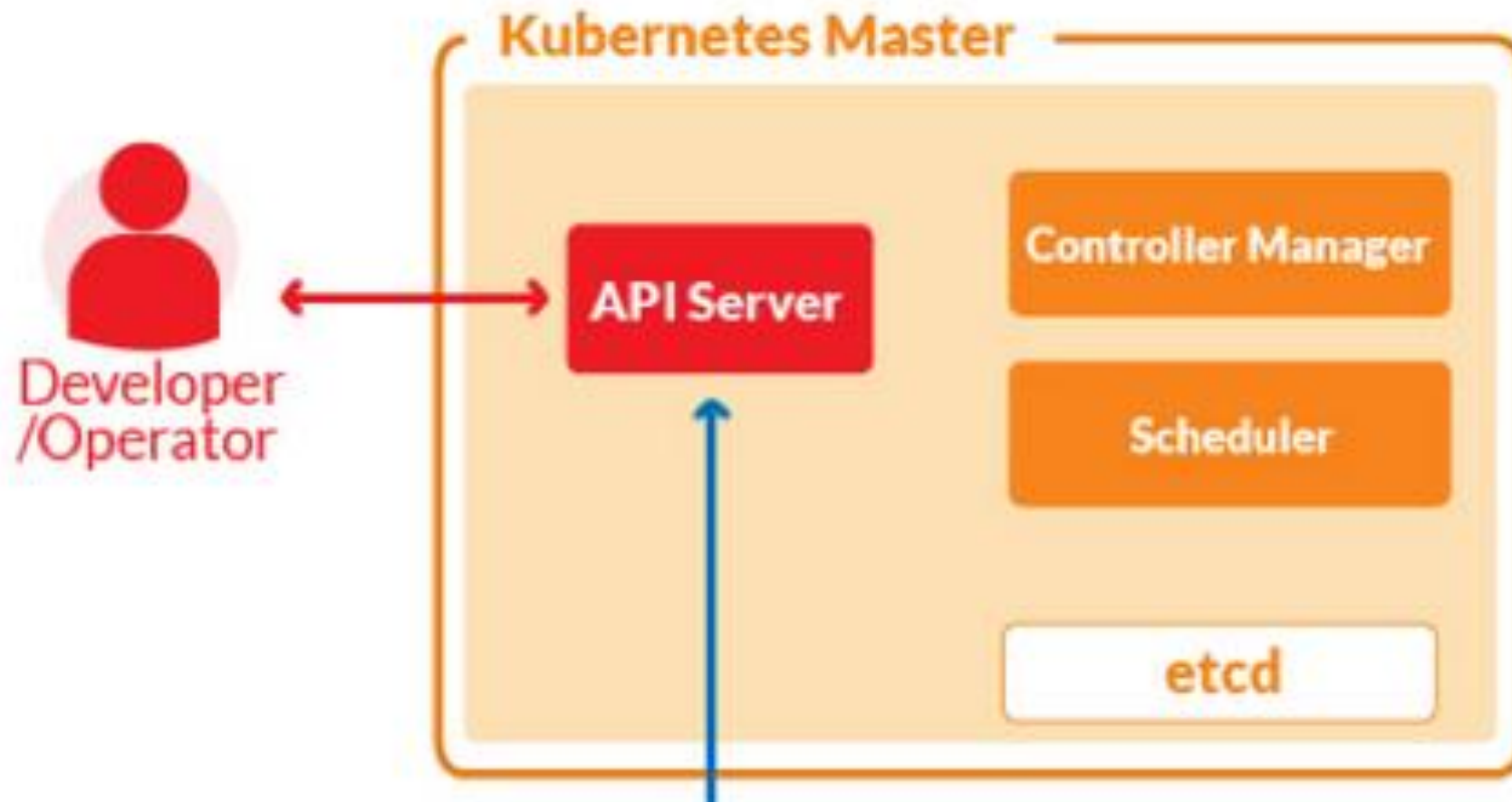
Que fait Kubernetes ? (2)

- Déclarer **l'architecture cible** et laisser le système travailler à **maintenir** cette cible
- Faire des mises à jour sans **interruption** de service
- Détecter les problèmes et **les résoudre tout seul**

Architecture

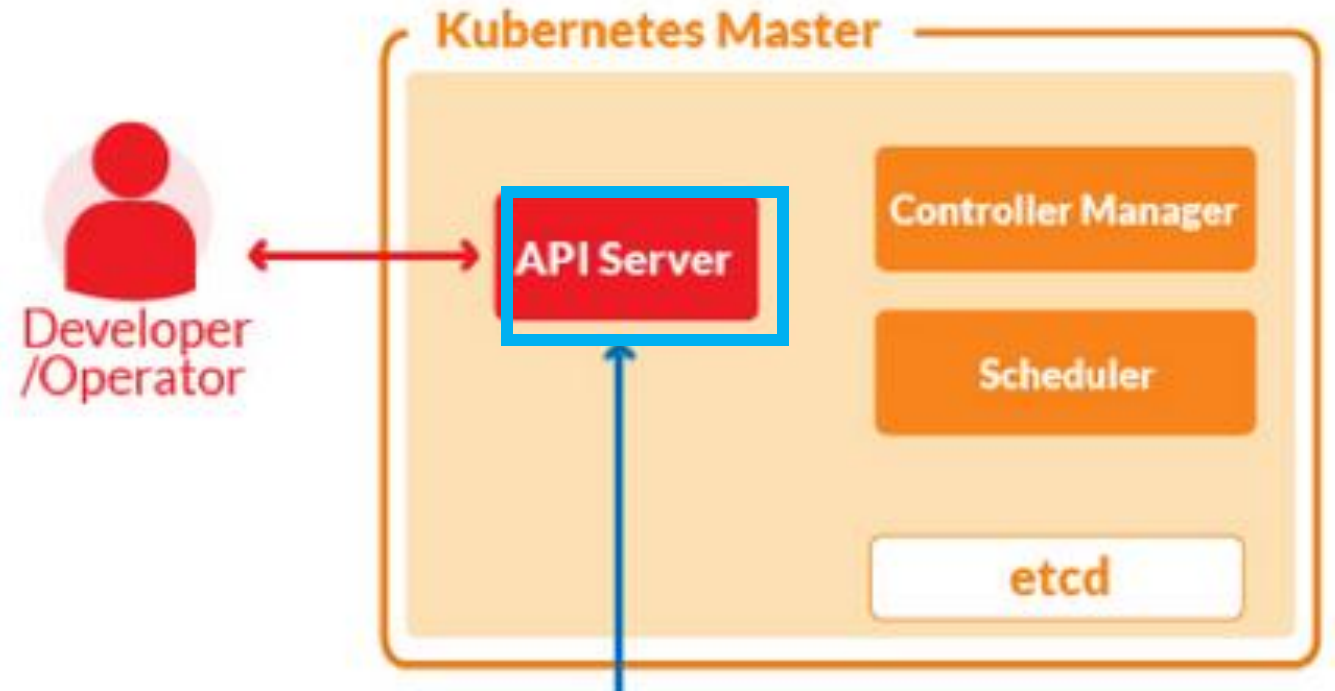


Principaux composants de master

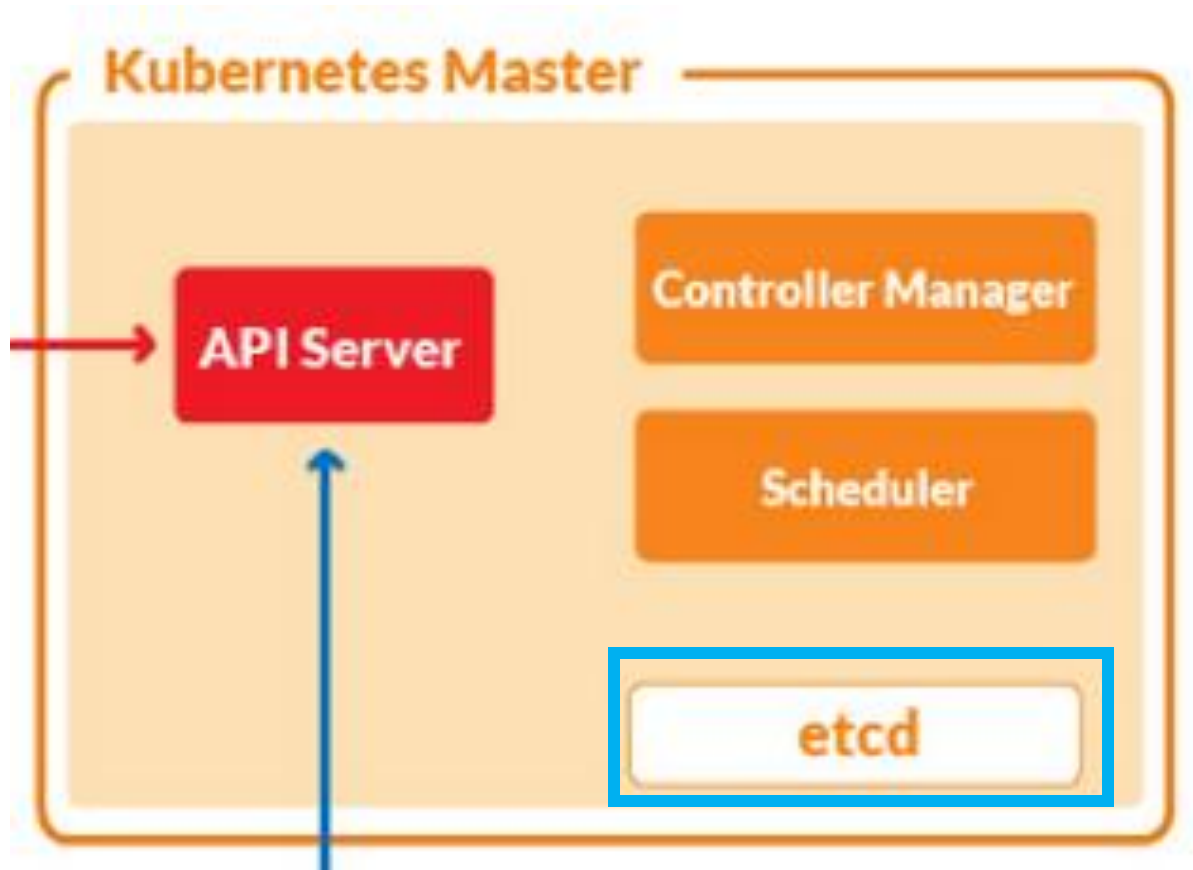


API Server

- Il fournit **l'interface** interne et externe
- Il gère et valide des requêtes REST
- Il permet aux clients de **configurer** la charge de travail et les containers

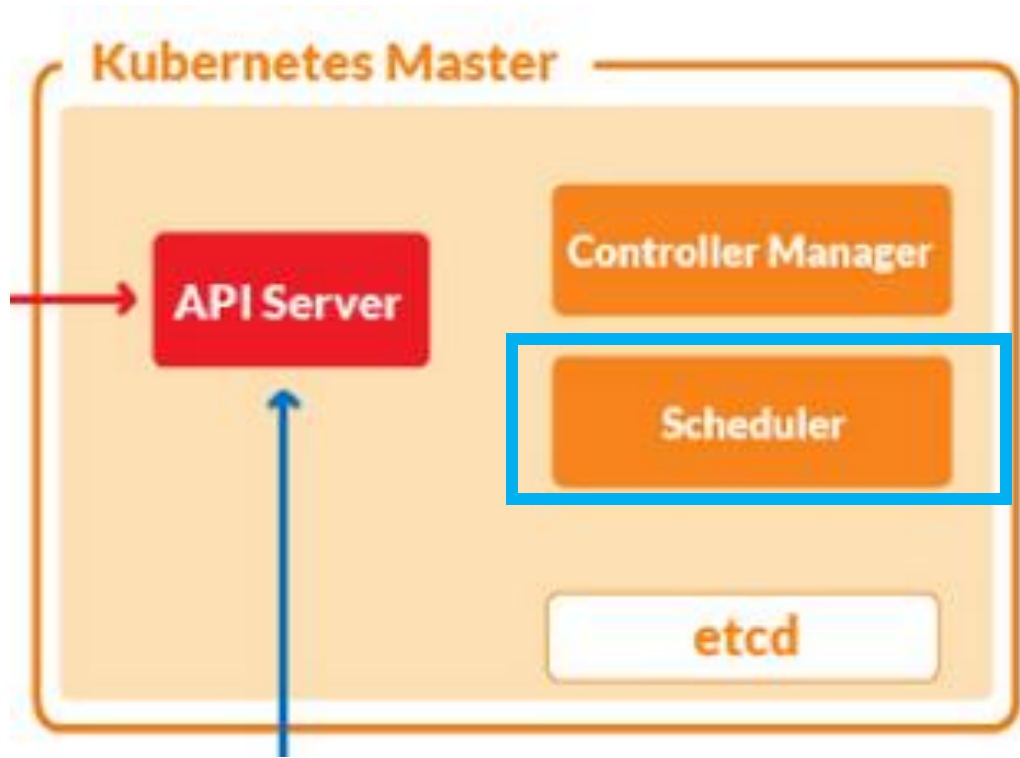


Etcd



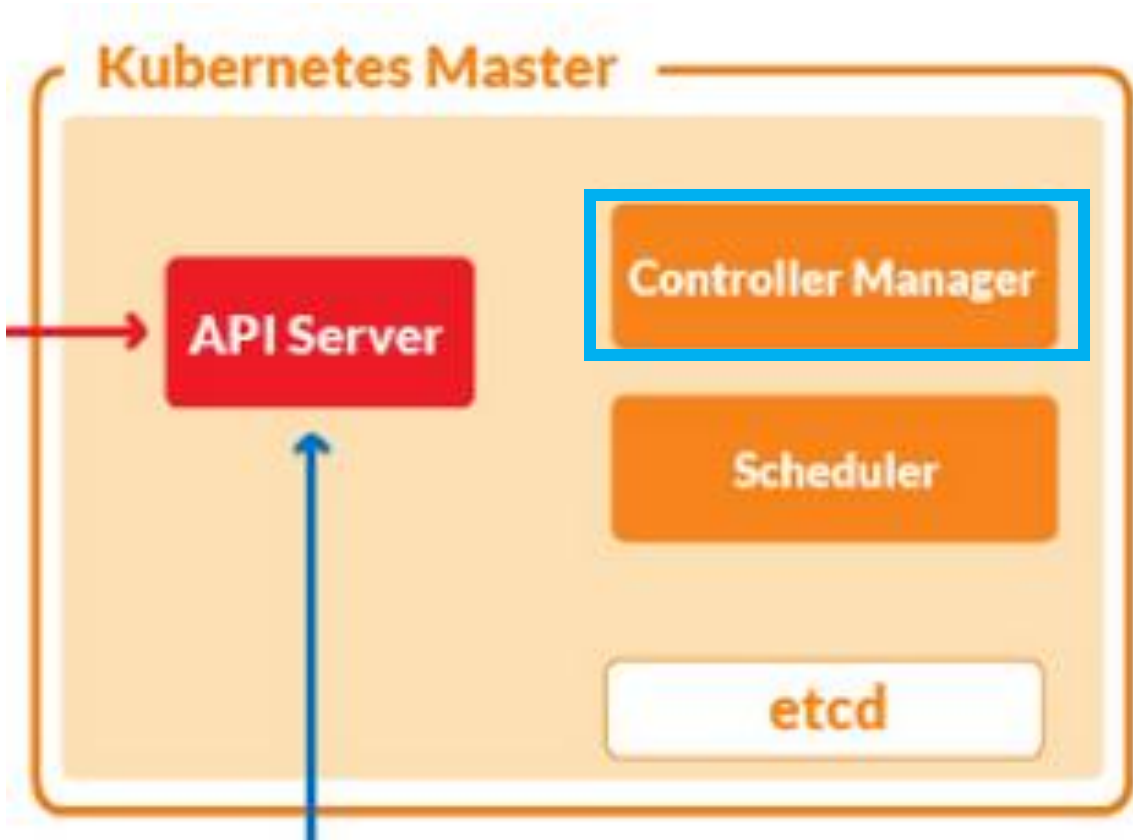
- Base de données distribuée de type **clé-valeur**
- **Légère**
- **Stocke** de manière fiable:
 - Les données de **configuration** du cluster
 - **L'état** du cluster

L'ordonnanceur (Scheduler)



- **Sélectionne** quel noeud devrait faire tourner un **pod**
- **Gère l'utilisation** des ressources sur chaque nœud
- L'ordonnanceur doit connaître:
 - les **ressources disponibles**
 - les **ressources assignées** sur les serveurs.

Gestionnaire de contrôle (Controller manager)

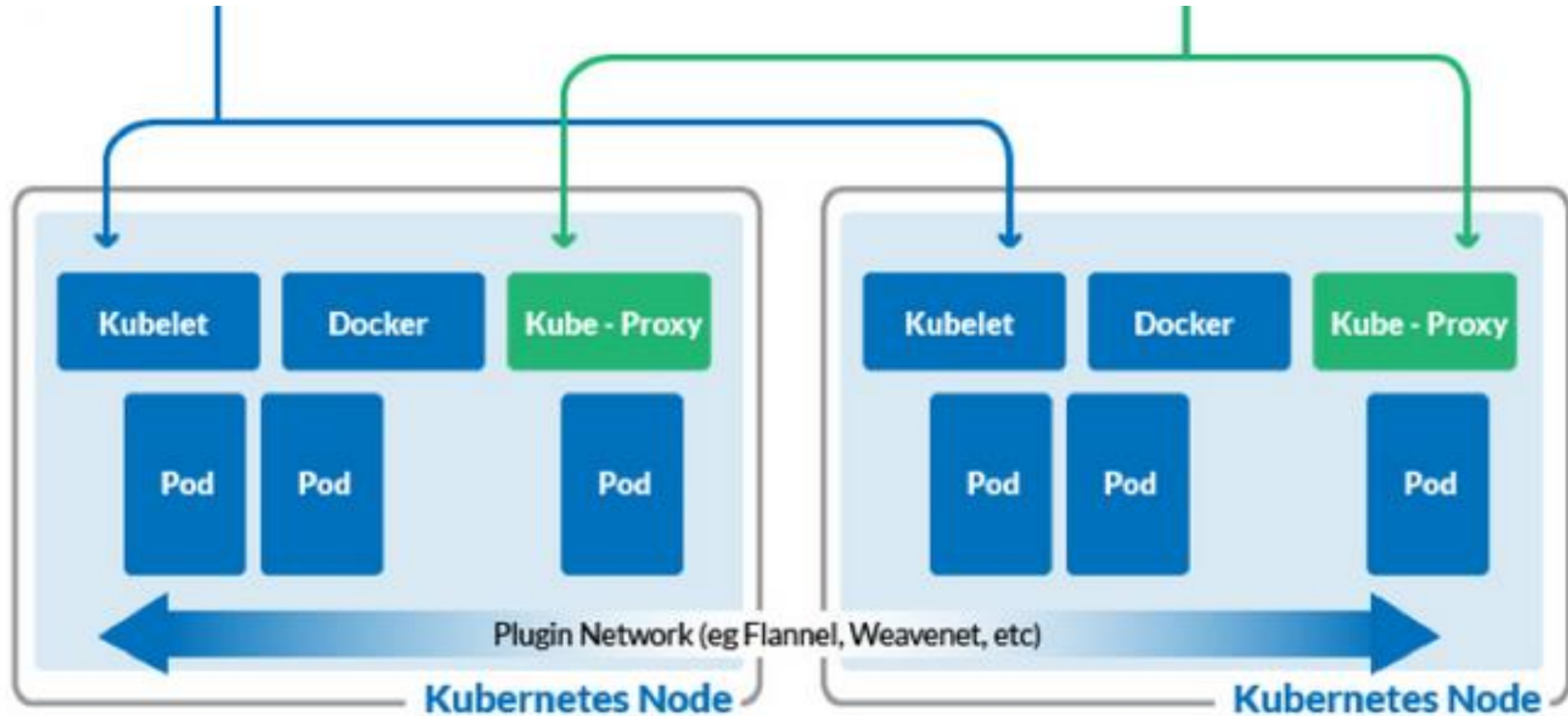


➤ Processus dans lequel s'exécutent les **contrôleurs** principaux :

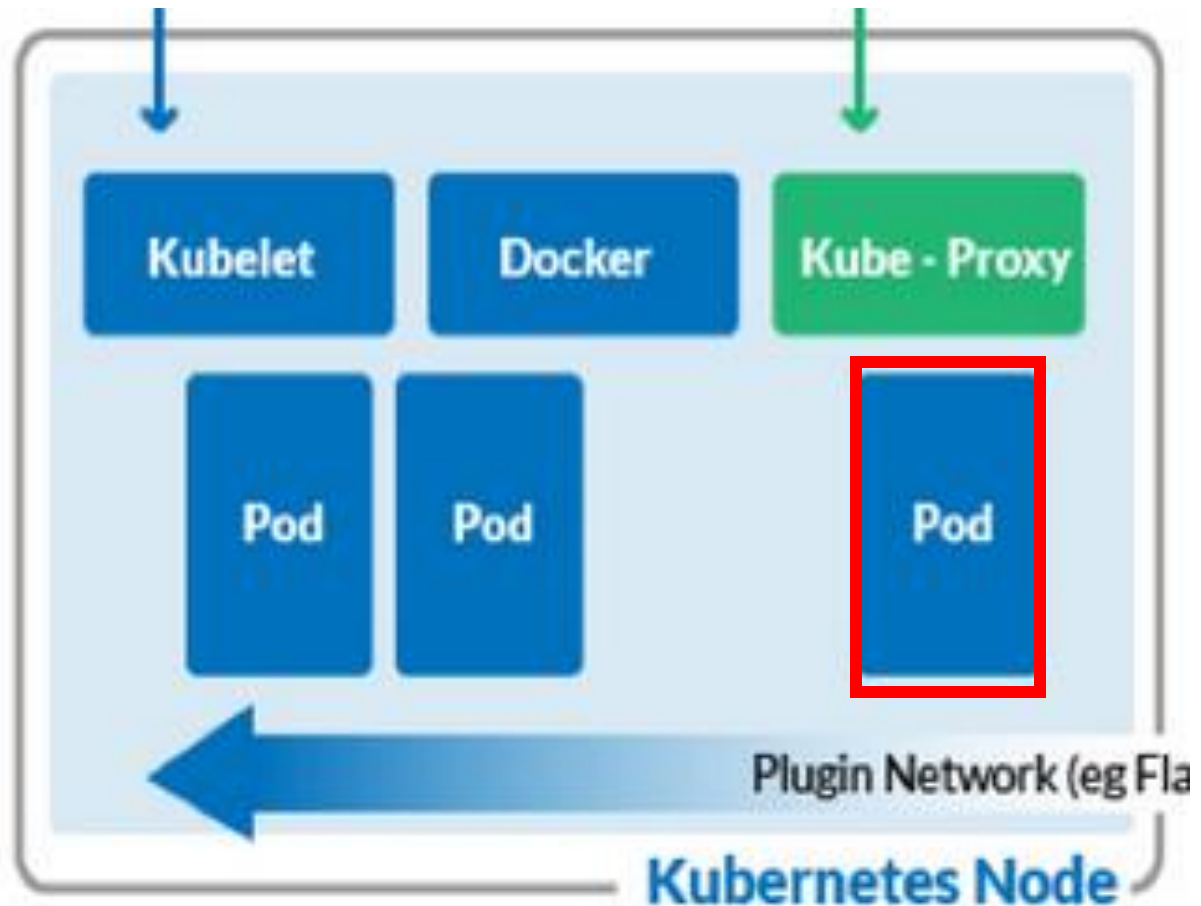
➤ Node controller (DementSet)

➤ Replication controller

Principaux composants de Nodes

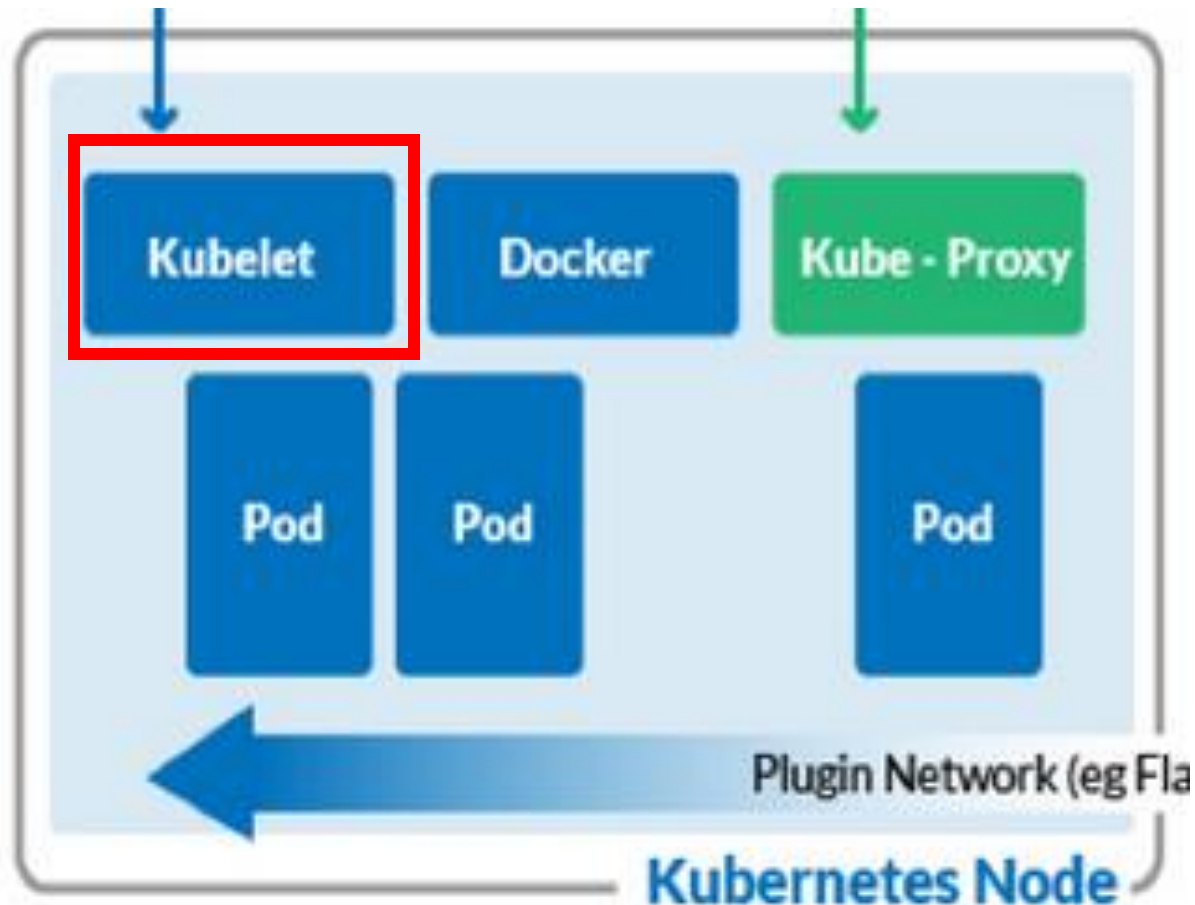


Pods



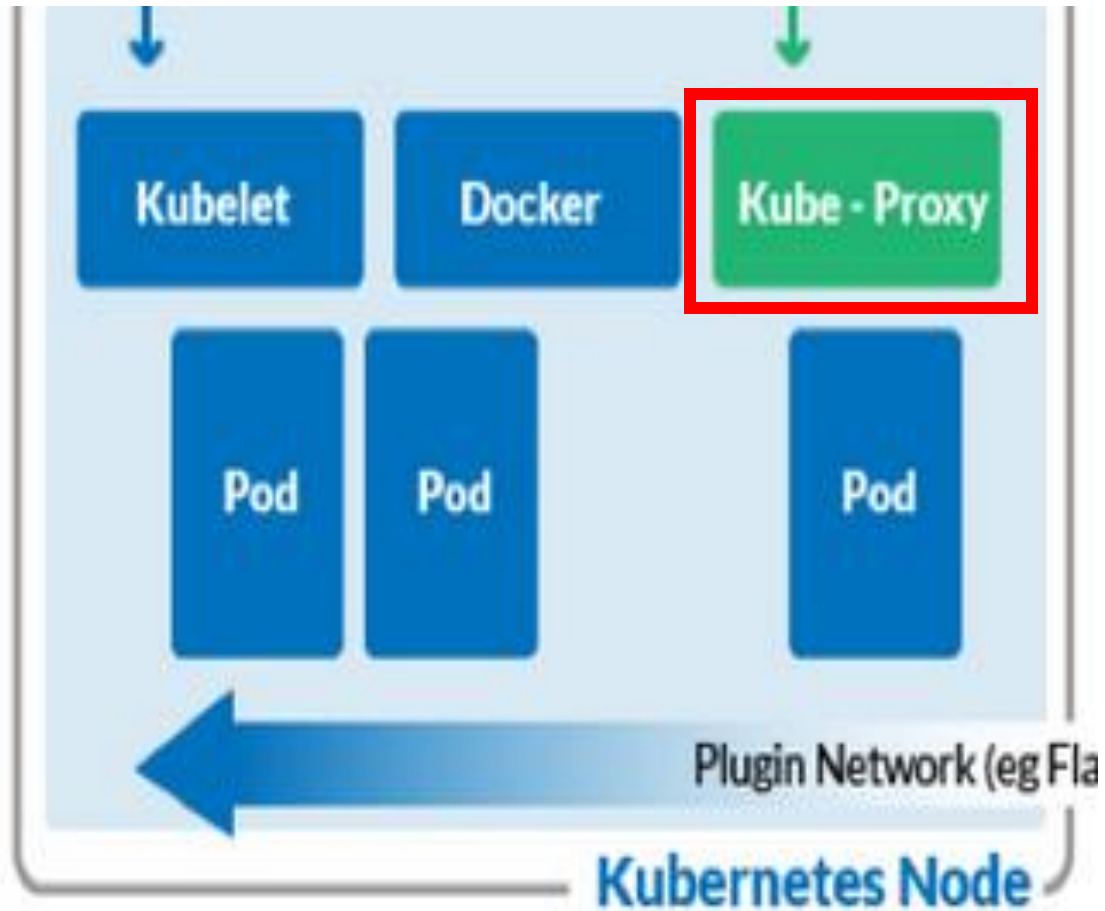
- Un ou plusieurs **conteneurs**
 - Partage le même **adresse(IP)**
- Exécuté sur la même **machine P/L**

Kubelet



- Un agent qui **s'exécute sur chaque nœud**
- **Envoie des reports** sur l'état des nœuds au master
- Le kubelet fonctionne en termes de **PodSpec**
 - Un **objet** YAML ou JSON
 - Il est **fourni** par le master
- **Garantit** que les conteneurs décrits
 - En cours d'exécution
 - en bonne santé

Kube-proxy



- **Proxy** réseau qui s'exécute sur chaque nœud
- Gère le **service** d'abstraction ainsi que d'autres opérations réseaux
- Responsable d'effectuer le **routing** du trafic vers le conteneur approprié
- Ex: communication entre les nœuds

cAdvisor

- Un agent qui :
 - **Surveille**
 - **Récupère** les données de consommation des ressources et des performances
 - Le taux d'utilisation CPU , RAM
 - l'utilisation disque et réseau des conteneurs

Installation de docker et de Kubernetes

- Installer minikube ou docker desktop
- `$> minikube start`
- `$> eval $(minikube docker-env)`
- Vérifier l'installation
 - `$> docker --version`
 - Docker version 19.03.5, build 633a0ea
 - `$> kubectl version`
 - Client Version: version.Info{Major:"1", Minor:"14", GitVersion:"v1.14.8", GitCommit:"211047e9a1922595eaa3a1127ed365e9299a6c23", GitTreeState:"clean", BuildDate:"2019-10-15T12:11:03Z", GoVersion:"go1.12.10", Compiler:"gc", Platform:"darwin/amd64"}
 - Server Version: version.Info{Major:"1", Minor:"14", GitVersion:"v1.14.8", GitCommit:"211047e9a1922595eaa3a1127ed365e9299a6c23", GitTreeState:"clean", BuildDate:"2019-10-15T12:02:12Z", GoVersion:"go1.12.10", Compiler:"gc", Platform:"linux/amd64"}

Créer une image docker

- `$> docker build . -t app-node-test:v1`

```
1 FROM node:12.14.0
2 EXPOSE 8080
3 COPY server.js .
4 CMD node server.js
```

```
1 var http = require('http');
2
3 var handleRequest = function(request, response) {
4   console.log('Received request for URL: ' + request.url);
5   response.writeHead(200);
6   response.end('Hello World!');
7 };
8 var www = http.createServer(handleRequest);
9 www.listen(8080);
```

Créer un pod Kubernetes

- `$> kubectl create deployment app-test --image=app-node-test:v1`
- `$> kubectl get deployments`
 - | NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|----------|-------|------------|-----------|-------|
| app-test | 1/1 | 1 | 1 | 2m36s |
- `$> kubectl get pods`
 - | NAME | READY | STATUS | RESTARTS | AGE |
|---------------------------|-------|---------|----------|-----|
| app-test-64874584f5-6p4l9 | 1/1 | Running | 0 | 47s |
- `kubectl delete pod "name"`

Créer un service Nodeport

- `$> kubectl expose deployment app-test --type=NodePort --port 8080`
 - service/app-test **exposed**
- `$> kubectl describe service app-test`
- `$> minikube service app-test`
- `$> kubectl delete service app-node`

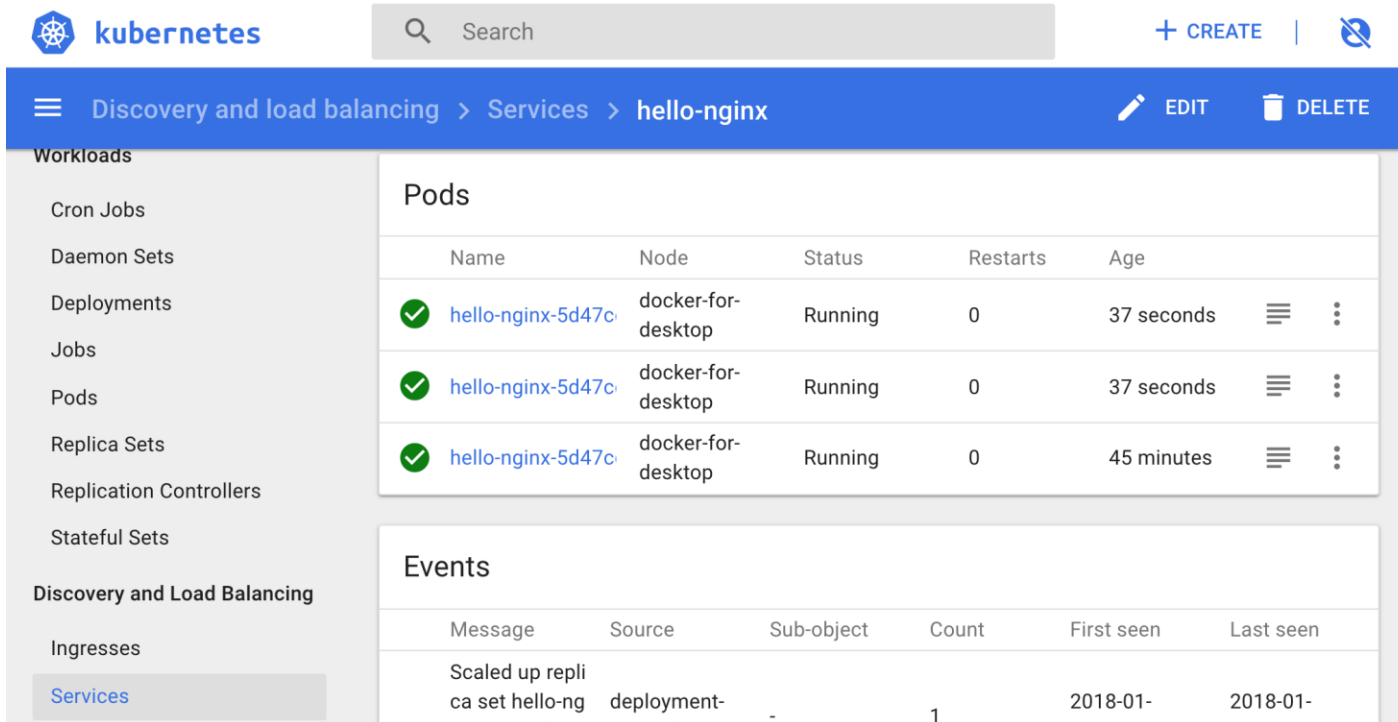
Possibilité de configurer un fichier yaml

application/deployment.yaml

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
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:1.7.9
          ports:
            - containerPort: 80
```

- L'exécution de Kubectl est limitée car nous ne pouvons pas spécifier toutes les spécifications comme options
- kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

Créer des réplicas



The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a search bar, and a '+ CREATE' button. The breadcrumb trail indicates the path: 'Discovery and load balancing > Services > hello-nginx'. The left sidebar lists various workload types, with 'Services' highlighted under the 'Discovery and Load Balancing' section. The main content area displays the 'Pods' table for the 'hello-nginx' service, showing three replicas in a 'Running' state on the 'docker-for-desktop' node. Below the pods table, an 'Events' table shows a 'Scaled up replica set hello-nginx-5d47c' event.

Pods						
	Name	Node	Status	Restarts	Age	
✓	hello-nginx-5d47c	docker-for-desktop	Running	0	37 seconds	⋮
✓	hello-nginx-5d47c	docker-for-desktop	Running	0	37 seconds	⋮
✓	hello-nginx-5d47c	docker-for-desktop	Running	0	45 minutes	⋮

Events						
	Message	Source	Sub-object	Count	First seen	Last seen
	Scaled up replica set hello-nginx-5d47c	deployment-controller	-	1	2018-01-10T10:40:10Z	2018-01-10T10:40:10Z

- `$> kubectl scale --replicas=3 deployment app-test`
 - deployment "app-test" scaled

Services exposés

☰

Discovery and load balancing > Services

Overview

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Ingresses

Services

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age	
✓ hello-nginx	run: hello-nginx	10.107.132.22	hello-nginx:80 hello-nginx:302	-	2 minutes	⋮
✓ kubernetes	component: kube-apiserver provider: kubernetes	10.96.0.1	kubernetes:443 kubernetes:0 T	-	8 hours	⋮

Conclusion

- Kubernetes est l'*avenir* du déploiement
 - Beaucoup d'entreprises ont déjà déménagé et certaines sont en transition
- Très bien documenté
- Totalelement adaptable et efficace
 - Peut être utilisé avec n'importe quel outil de containerisation
 - "Je veux 2 instances de ma page Web accessible aux utilisateurs, 3 instances de mon service de catalogue et 10 instances de mon service de panier d'achat" et Kubernetes le fait

References

- <https://fr.wikipedia.org/wiki/Kubernetes>
- <https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams>
- <https://kubernetes.io/fr/docs/concepts/overview/components/>
- <https://www.youtube.com/watch?v=KV37yqCtpGY>
- https://www.youtube.com/watch?v=zztKO0iRX_w

Credits

- <https://containerjournal.com/topics/container-ecosystems/kubernetes-vs-docker-a-primer/>
- <https://pngriver.com/download-container-png-file-for-designing-projects-40903/>
- <https://www.stickpng.com/fr/img/icones-logos-emojis/societes-de-technologie/logo-kubernetes>