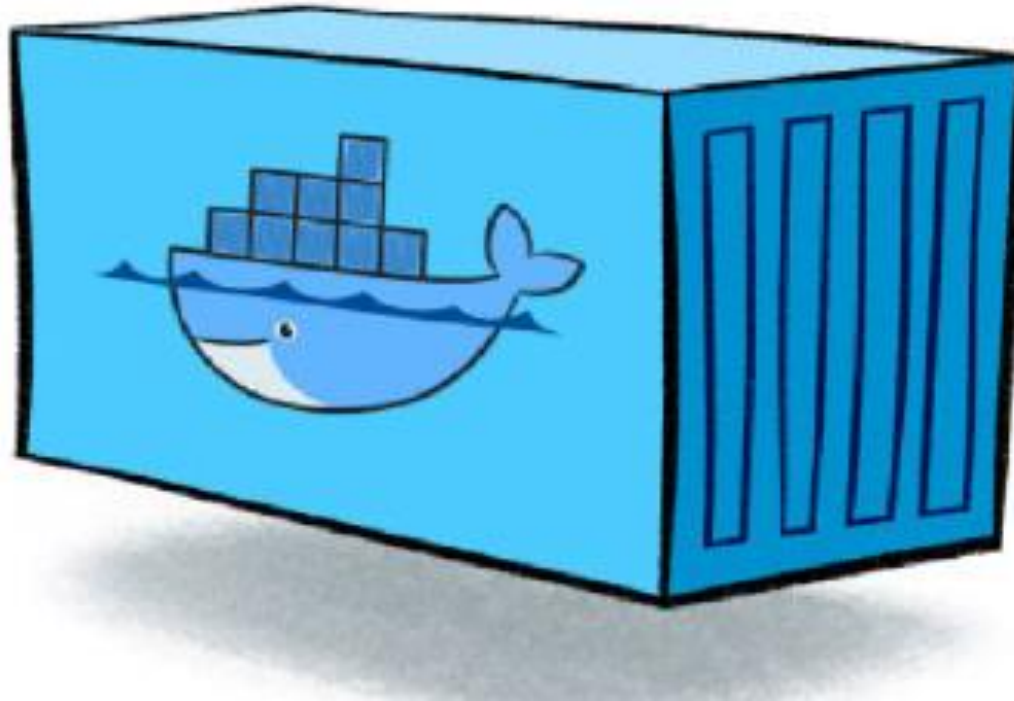




kubernetes

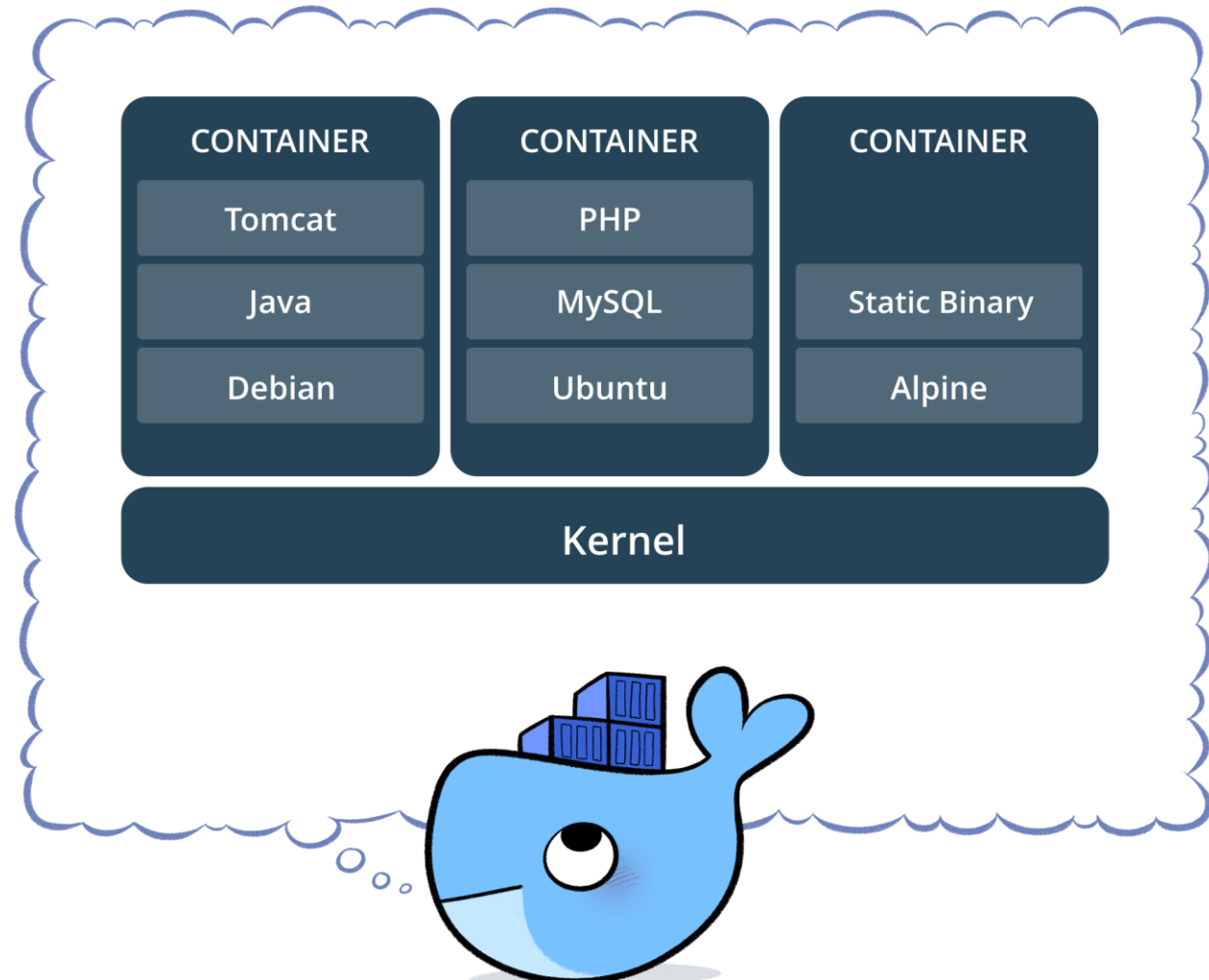
Orchestrator de containers

C'est quoi un container ?

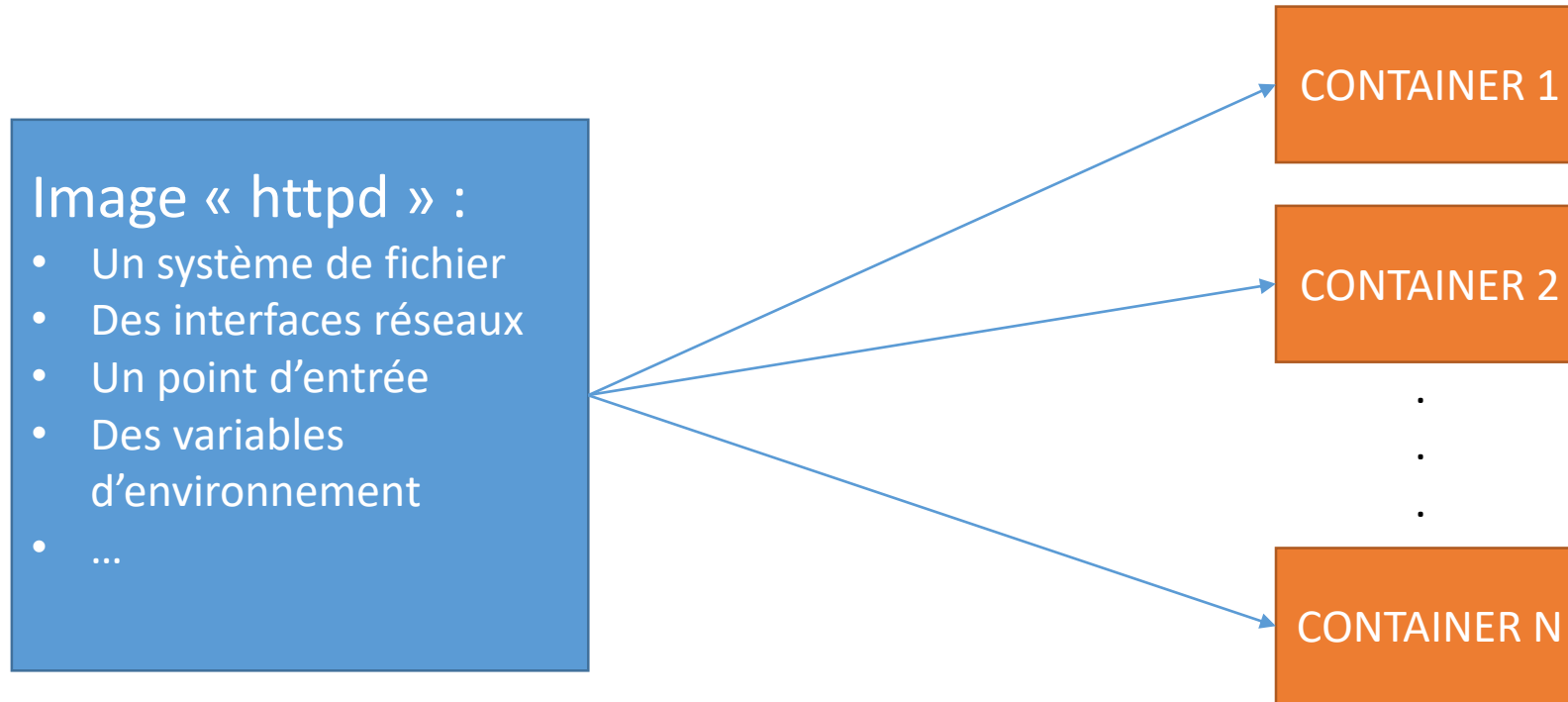


Environnement isolé

- Processus isolés
- Système fichiers isolé
- Réseau isolé



Les images docker



Les images Docker

```
# A basic apache server. To use either add or bind mount content under /var/www
FROM ubuntu:12.04

MAINTAINER Kimbro Staken version: 0.1

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

Un Joomla en 30 secondes avec Docker

- `docker run -ti --name mysql -e MYSQL_ROOT_PASSWORD=admin -p 3306:3306 -d mysql:5.7`
- `docker run --name joomla --link mysql:mysql -p 80:80 -d joomla`

100 Joomla en 30s avec Docker ?



100 Joomla en 30s avec Docker ?

- Des containers sur plusieurs serveurs
 - Une gestion du stockage
 - Une gestion du réseau et des URL
 - Un redémarrage du container en cas de problème
 - Une gestion des ressources (CPU/RAM)
-
- Du load-balancing ?
 - De l'isolation réseau ?



kubernetes

K8S

- Issu du projet Borg de Google
- Projet de la Cloud Native Computing Foundation (CNCF)
- 2 en 1 : gestion du cluster et gestion des container

- Première release en juillet 2015
- Une release tous les 3 mois
- 1.11 en juin 2018

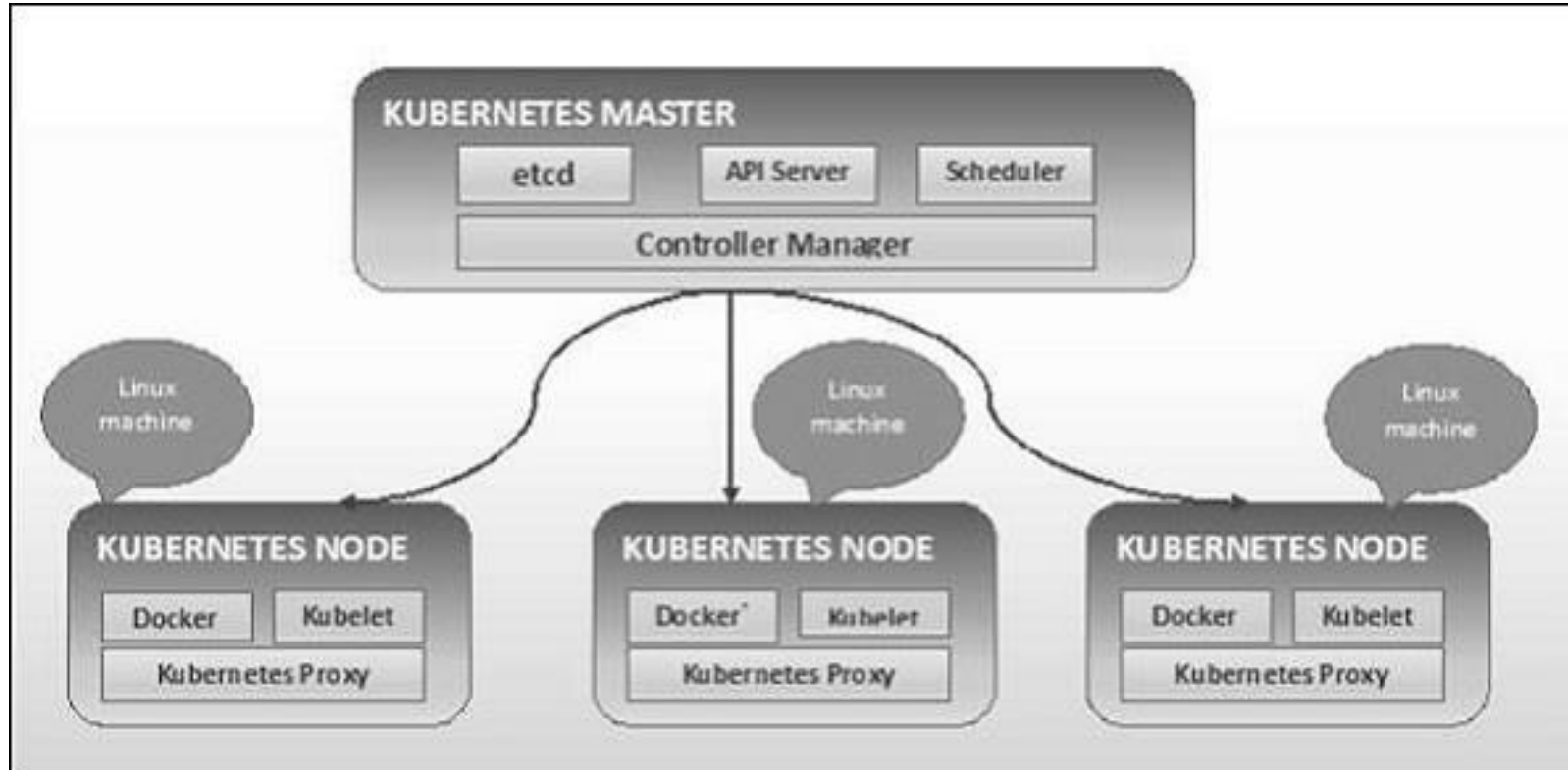
Cluster management

- Ajout / suppression de node
- Mise en maintenance
- Monitoring
- Gestion des ressources matérielles

Container scheduler

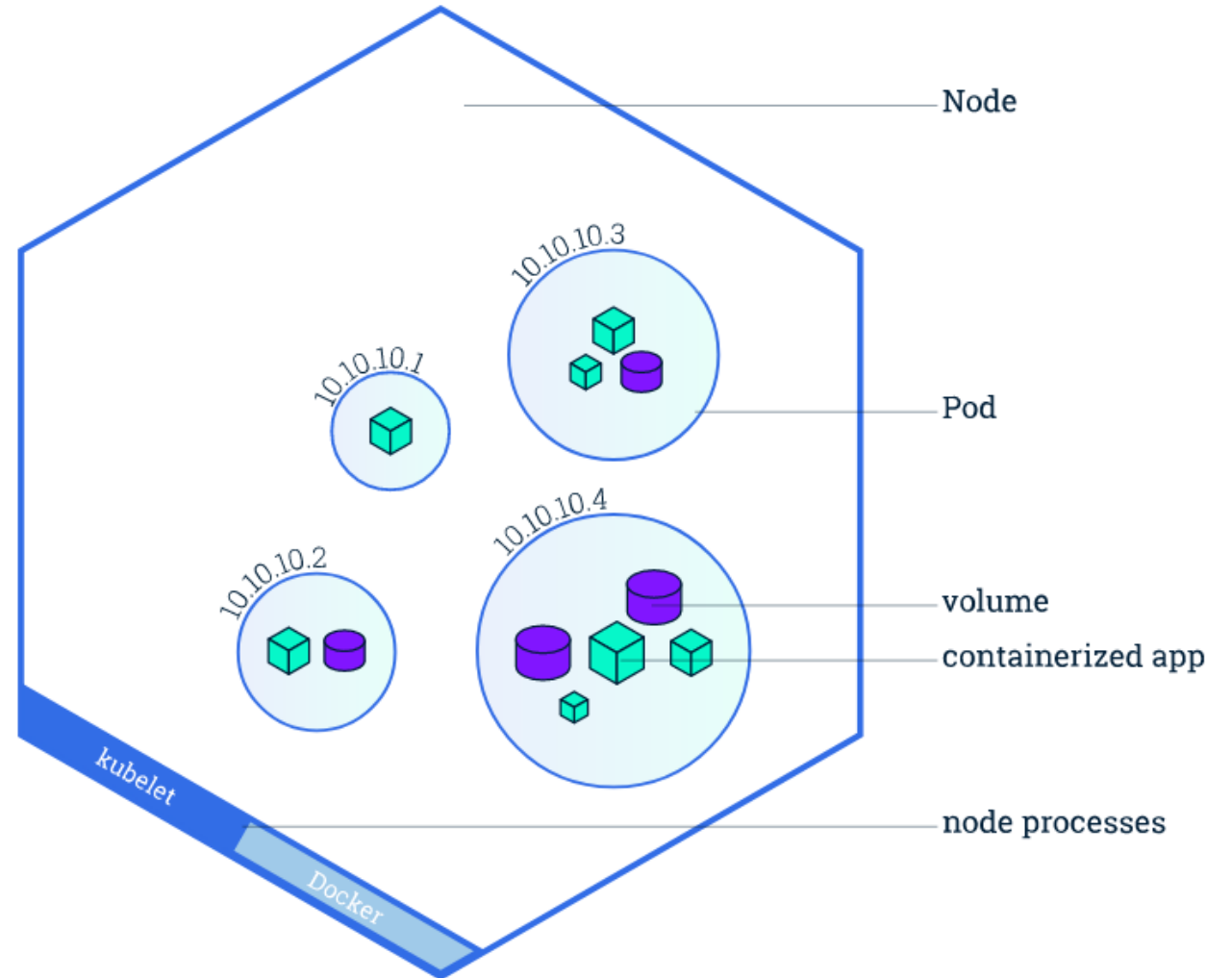
- Placement du container
- Health check
- Scalabilité
- Load-balancing
- Gestion du stockage

Architecture



Le Pod

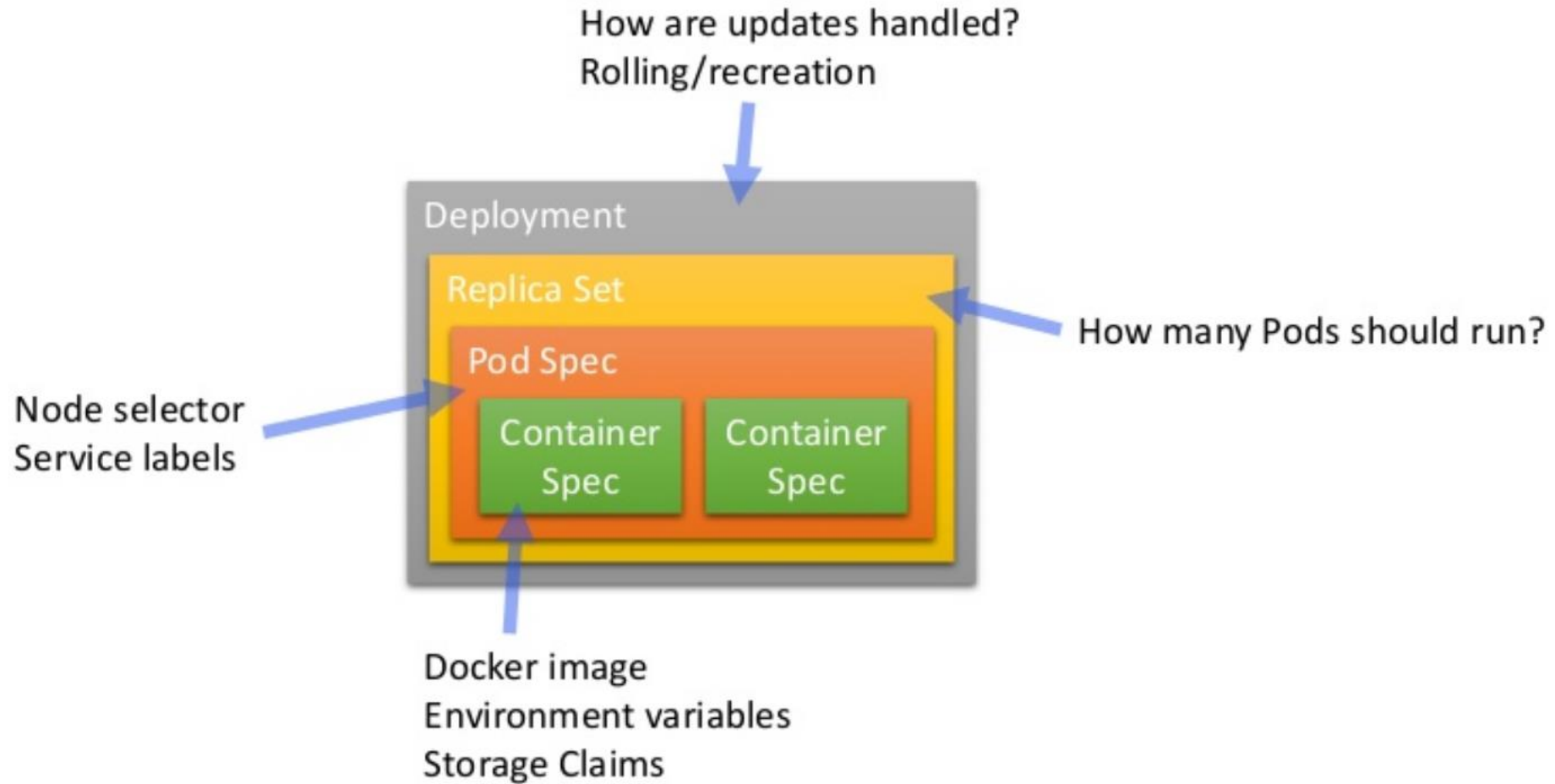
- Plus petite unité
- Ensemble de container
- Ephémère
- Partage les volumes
- Indivisible



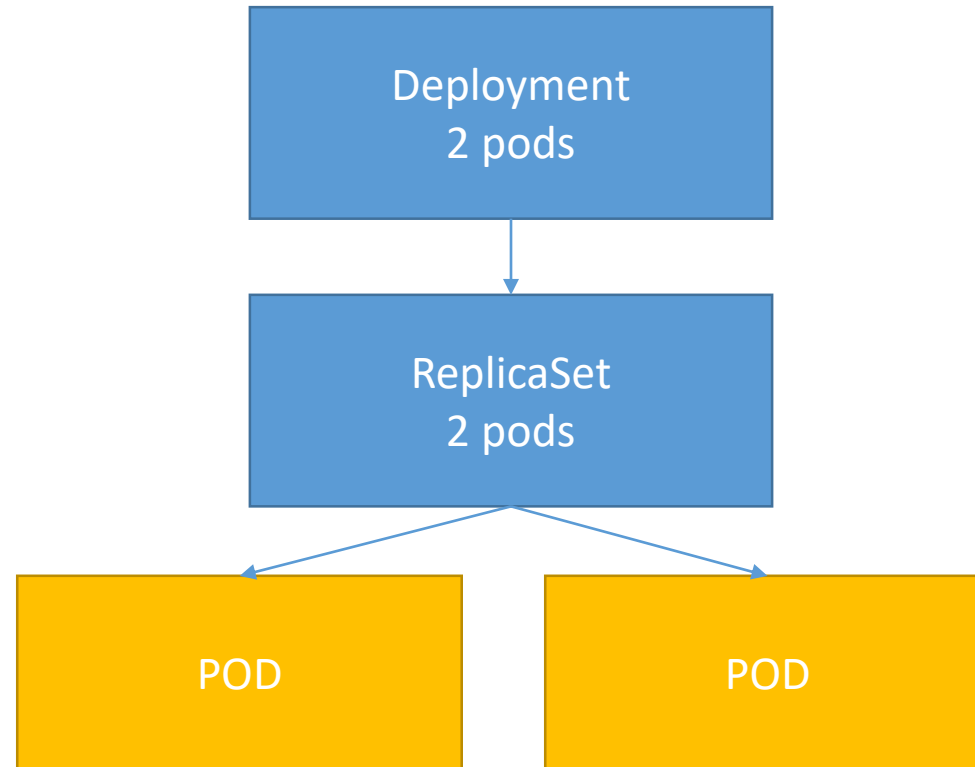
Exemple

```
kind: Pod
apiVersion: v1
metadata:
  name: example-01-pod
spec:
  containers:
  - name: writer
    image: busybox
    command: ['sh', '-c', 'i=0 ; while true; do echo $i > /mnt/index.html; i=$((i+1)); sleep 1; done']
    volumeMounts:
    - mountPath: "/mnt"
      name: storage
  - name: reader
    image: busybox
    command: ['sh', '-c', 'while true; do cat /mnt/index.html; sleep 1; done']
    volumeMounts:
    - mountPath: "/mnt"
      name: storage
  volumes:
  - name: storage
    emptyDir: {}
```

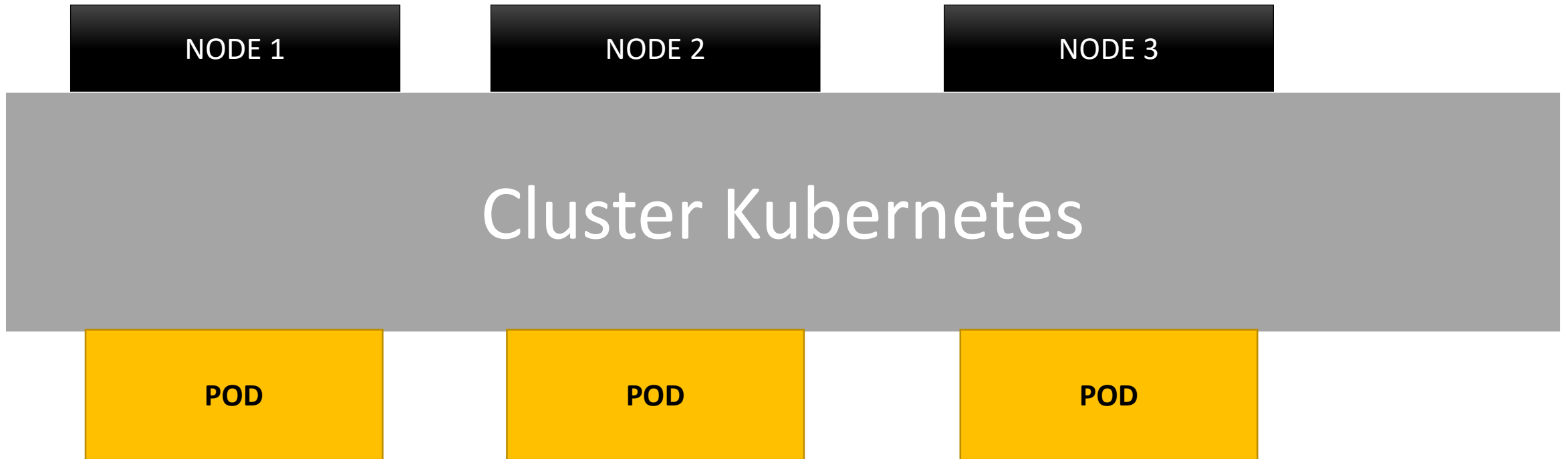
ReplicaSet et Deployment



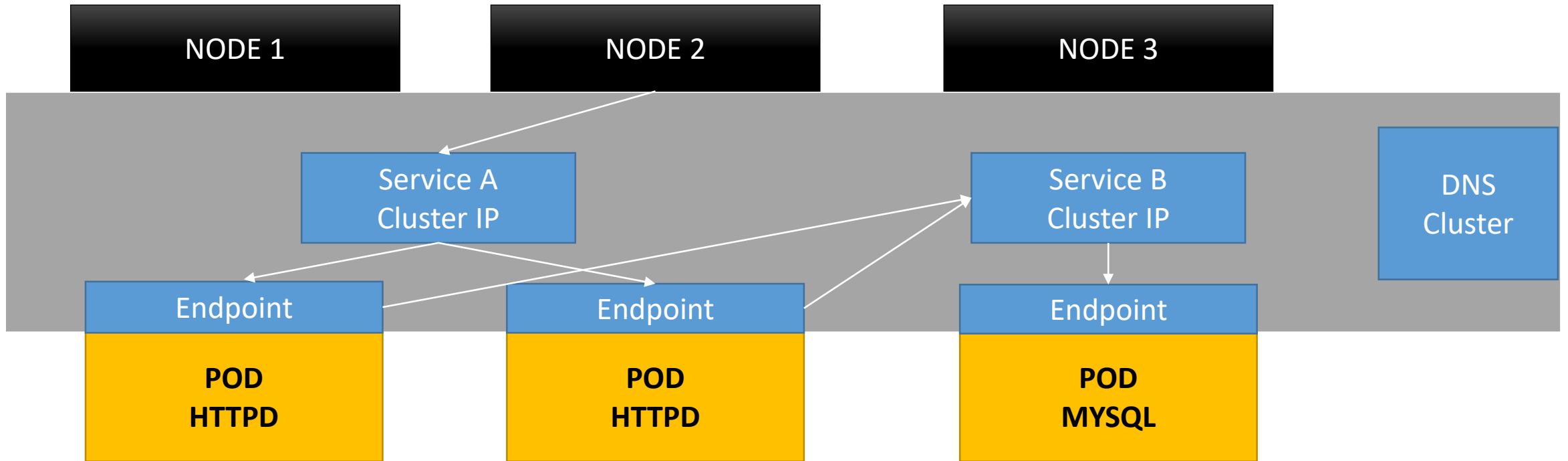
Exemple



Exposer des pods : les services



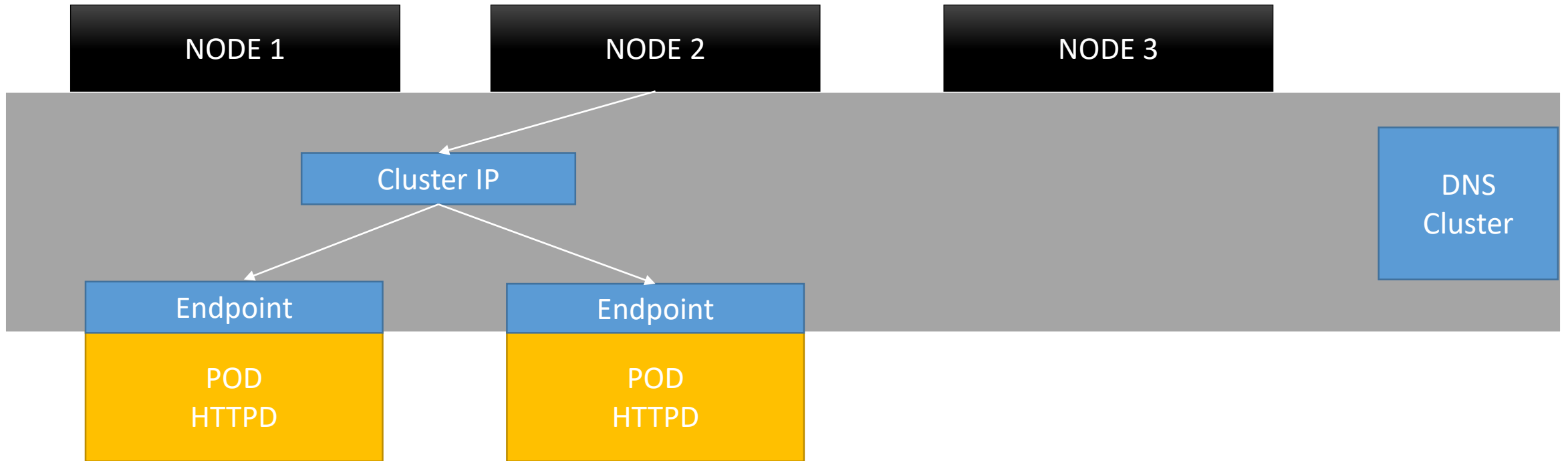
Le service



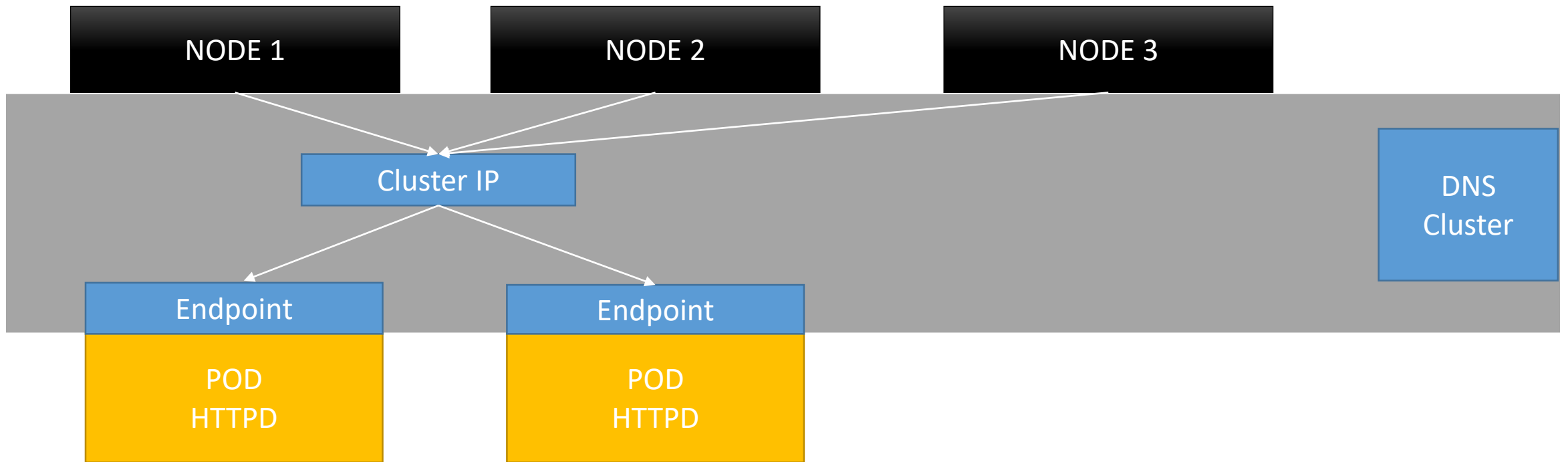
Le service

- Permet d'exposer des pods en interne et en externe
- Crée 1 endpoint par pod
- 0/1 clusterIP + 1 entrée DNS
- 3 types de services :
 - ClusterIP
 - NodePort
 - LoadBalancer

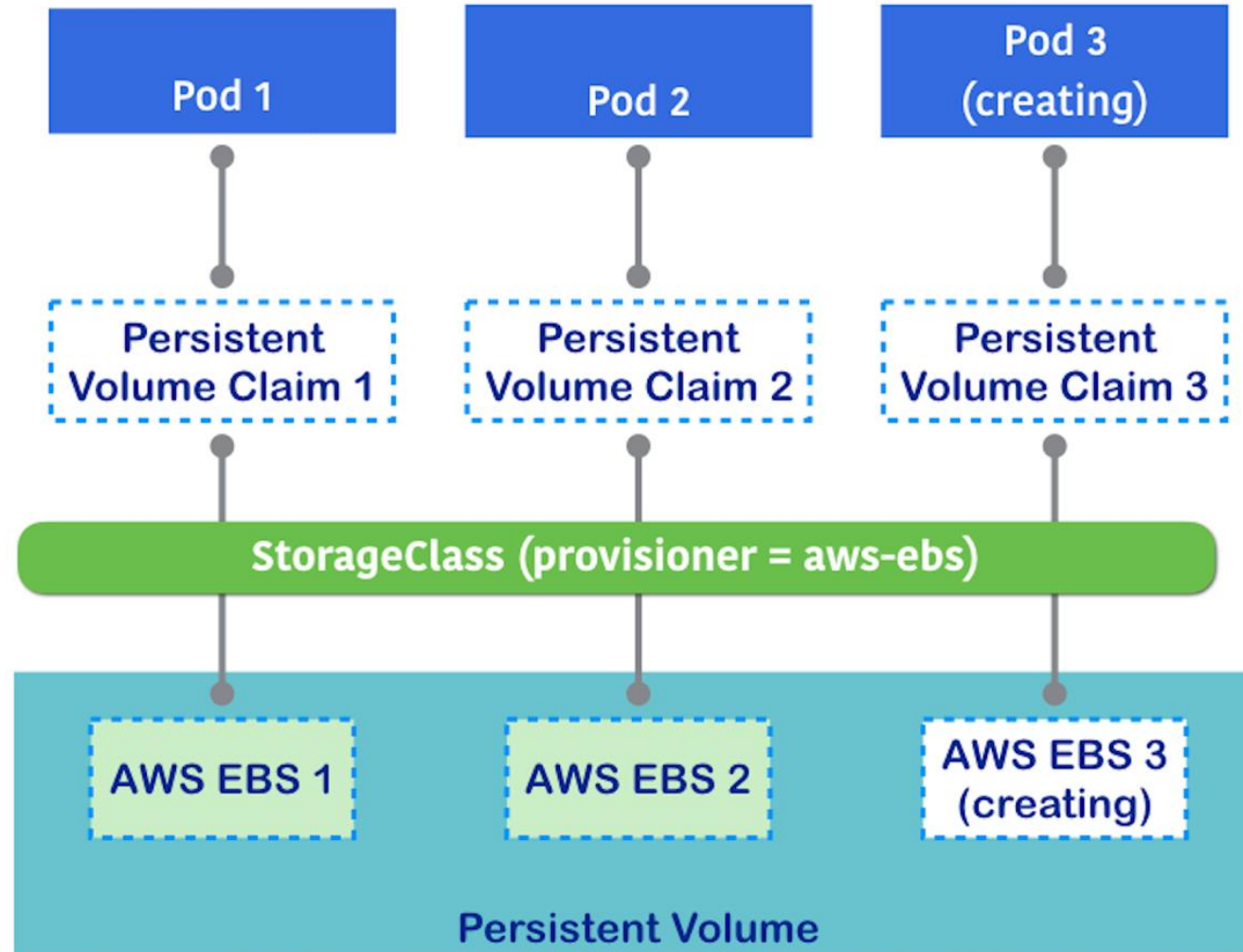
Exemple 1 : ClusterIP



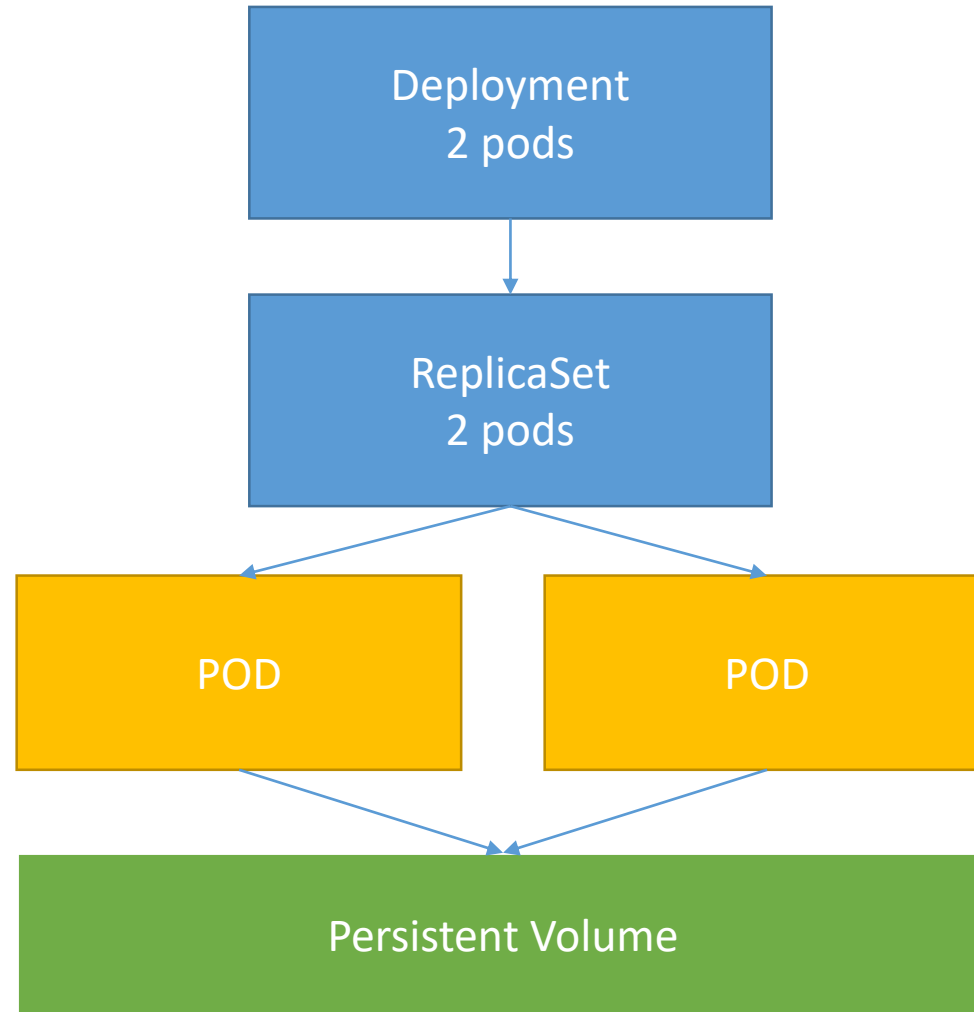
Exemple 2 : NodePort



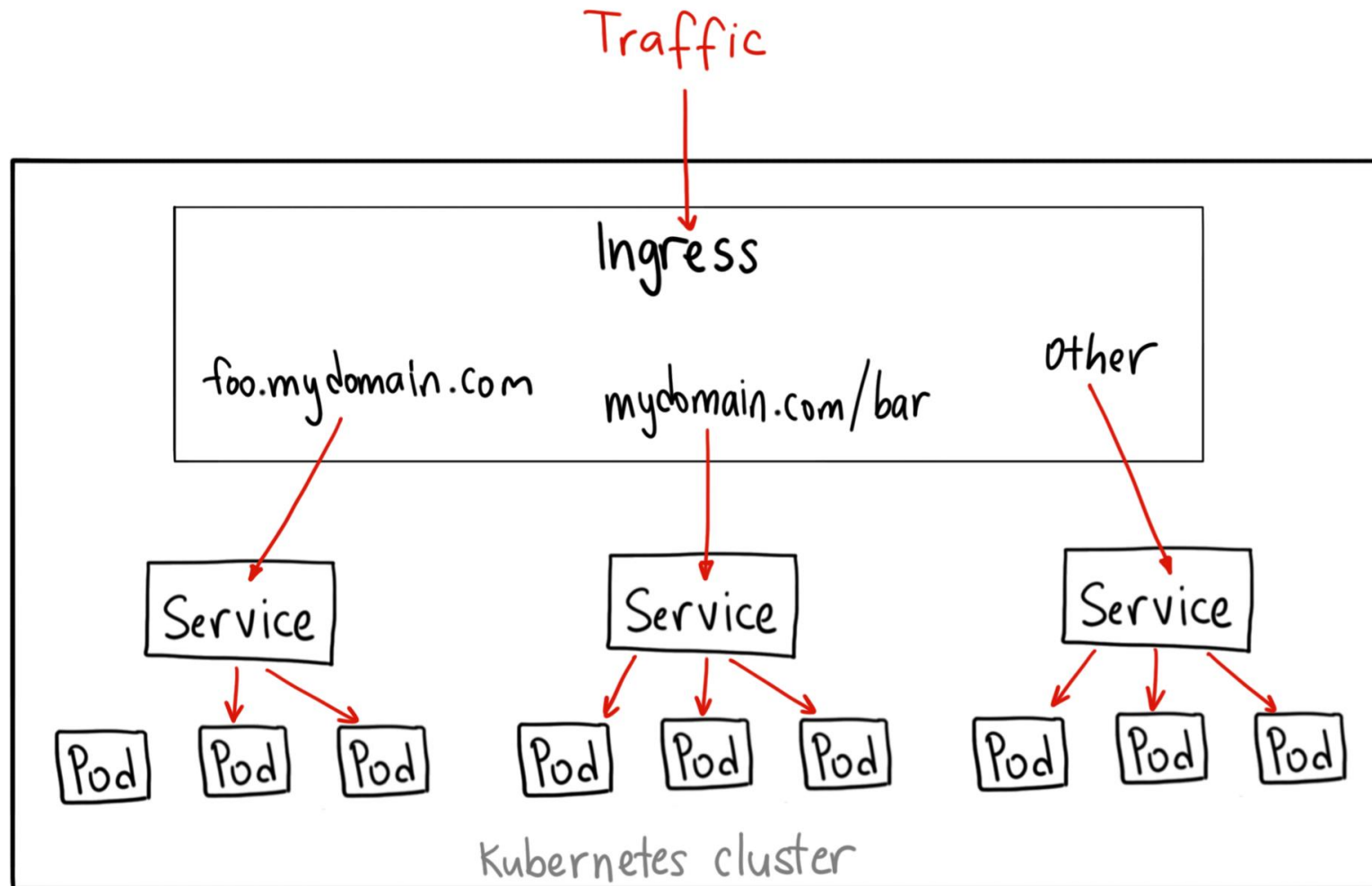
Stockage persistant



Exemple



Routage HTTP : Ingress



Routage HTTP : Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test-ingress
spec:
  rules:
  - host: www.timactive.com
    http:
      paths:
      - backend:
          serviceName: timactive-prestashop-apache-svc
          servicePort: 80
    tls:
      - hosts:
          - www.timactive.com
        secretName: timactive-com-tls
```

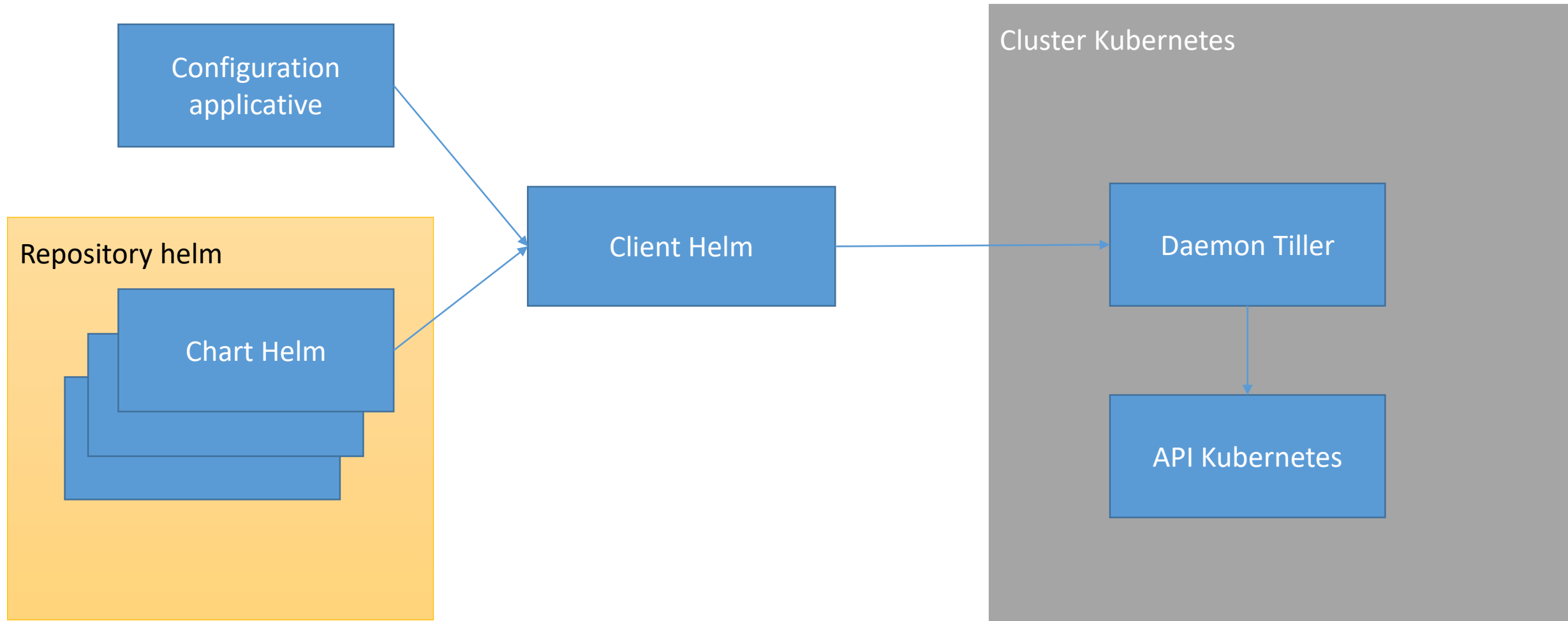
Et plein d'autres ressources...

- Pour les pod « statefull » : Statefulset
- Pour n'exécuter qu'une seule fois : Job
- Pour croner une tâche : CronJob
- Pour les configurations : ConfigMap
- Pour les secrets : Secret
- Pour le filtrage réseau : NetworkPolicy
- Pour sécuriser l'accès aux API : ServiceAccount, Role, RoleBinding
- Les quota ? Limitrange, Resourcesquotas
- ...

Et mes 100 Joomla ??



Déployer facilement : Helm



Joomla ? Trop facile !

```
joomlaUsername: user
joomlaPassword: toto
joomlaEmail: user@example.com

service:
  type: ClusterIP

ingress:
  enabled: true
  hosts:
    - name: joomla.local
      |   tls: false
      |   tlsSecret: joomla.local-tls

persistence:
  enabled: true
  apache:
    |   accessMode: ReadWriteOnce
    |   size: 1Gi
  joomla:
    |   accessMode: ReadWriteOnce
    |   size: 8Gi
```

Exemple : déployer Joomla

- Installation :

```
helm install --name joomla1 --f config.yaml stable/joomla
```

- Mise à jour :

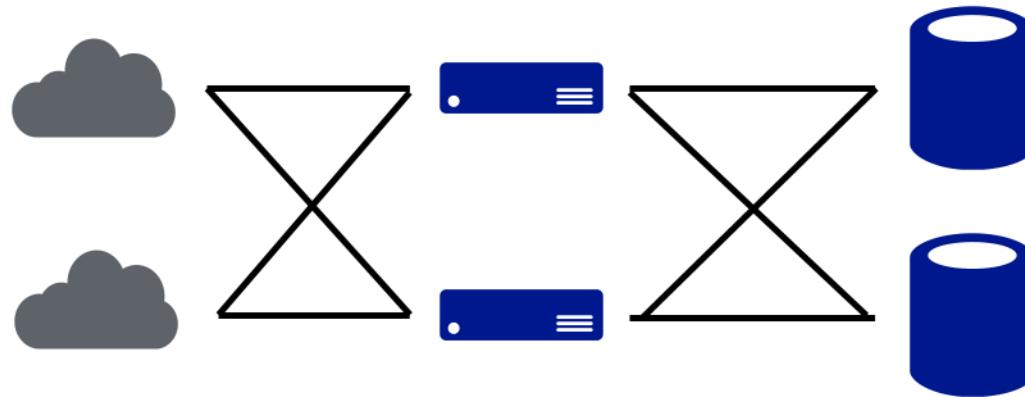
```
helm upgrade joomla1 --f config.yaml stable/joomla
```

- Suppression :

```
helm delete --purge joomla1
```

Bonus : Joomla en haute disponibilité

High Availability = System with No Single Point of Failure



Bilan

- Kubernetes offre :
 - La HA
 - La résilience aux pannes
 - Une gestion simple et efficace des applis
 - La gestion du réseau et du stockage

- Mais n'offre pas :
 - Le système de stockage
 - La HA réseau
 - La HA applicative
 - Le monitoring et la centralisation des logs

Questions