

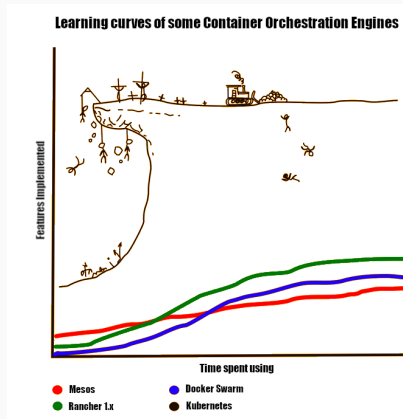
INTRODUCTION À KUBERNETES

Rémi Cailletaud

14 octobre 2024

ANF Mathrice 2024

La réputation d'un système complexe... est-elle méritée ?



Source: r/kubernetes

- PyCon 2013... Tout juste 10 ans !
- *Build once, run anywhere* : packaging de l'application et de ses dépendances. Runtime léger, magasin d'applications !
- *Separation of Concerns* : séparation de la logique métier — dans le conteneur de l'exploitation — logging, monitoring, réseau...
- Normalisation du format (OCI Image Format) et des runtimes (OCI Runtime) : containerd, cri-o, kata...

- Deux projets Google, dont Google Borg, en Java.
- Réécriture en Go, version 1.0 en 2015.
- Pilotage par Google, puis par la Cloud Native Foundation (Linux Foundation) depuis août 2018.

«We must treat the datacenter itself as one massive warehouse-scale computer.»

in The Datacenter as a Computer : An Introduction to the Design of Warehouse-Scale Machines

Luiz André Barroso Jimmy Clidaras, Urs Hölzle

- L'abstraction des couches matérielles et système.
- Le couplage faible des composants.
- Un surcoût minimal.
- Fonctionnement indifférent sur machines physiques et sur machine virtuelles.
- L'OS du cloud

Une API déclarative

On définit des «contrats» pour nos applications.

Des capacités d'autoréparation

Avant



Jonathan Schaeffer 9 h 41

Salut Rémi !

Quand tu pourras, redémarre influxdb un petit coup, j'ai fait une trop grosse requête !



Jonathan Schaeffer 10 h 07

rémi, je crois qu'il faut encore rebooter influxdb



Rémi Cailletaud 10 h 07

héhé

Après



Rémi Cailletaud 10 h 10

Lance-moi une instance d'InfluxDB.



Kubernetes 10 h 10

OK, je m'en occupe.



Kubernetes 10 h 25

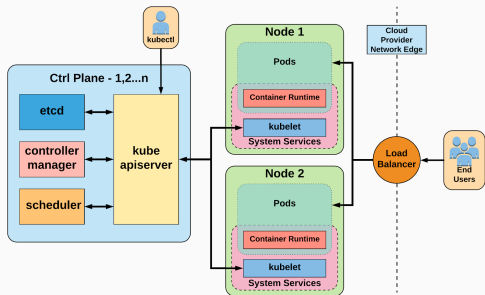
Tiens, Jonathan a encore lancé une monstro-requête. Je redémarre l'instance.

Une infrastructure immutable

- Tests facilités.
- Passage à l'échelle.
- Mise à jour et retour en arrière.

KUBERNETES : ARCHITECTURE

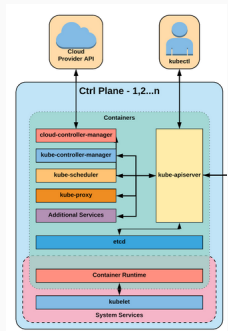
- Une séparation nette du plan de contrôle et du plan de travail.
- Configuration par un point unique, via l'API.
- Composants faiblement couplés : communication uniquement avec l'API.



Source: Introduction to Kubernetes by Rishabh Indoria (March 2019)

KUBERNETES : LE PLAN DE CONTRÔLE

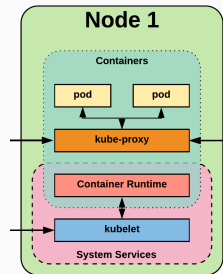
- L'API au centre du système.
- *etcd* pour le stockage clé/valeur.
- Des boucles de contrôles : *kube-controller-manager*.
- Un ordonnanceur : *kube-scheduler*.
- Un composant pour l'intégration au cloud : *cloud-controller-manager*.



Source: Introduction to Kubernetes by Rishabh Indoria (March 2019)

KUBERNETES : LE PLAN DE TRAVAIL

- Gestion des charges de travail (pods) : *kubelet*.
- Gestion des règles de forwarding et de l'équilibrage de charge : *kube-proxy* (ou un CNI l'implémentant)
- Un runtime compatible Container Runtime Interface : Docker, containerd, CRI-O, Kata...



Source: Introduction to Kubernetes by Rishabh Indoria (March 2019)

```
apiVersion: v1 # version de l'API
kind: Namespace # type de l'objet
metadata:
  name: anf24 # nom de l'objet
```

Ressource éphémère : Nom et IP non persistants !

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: test-ns
spec:
  containers:
  - image: nginxinc/nginx-unprivileged:latest # Image utilisée
    imagePullPolicy: Always
    name: nginx-pod
    containerPort: 80 # Port exposé en interne (purement informationnel)
    resources: # Ressources demandées et limites
      limits:
        cpu: 600m
        memory: 512Mi
      requests:
        cpu: 100m
        memory: 512Mi
```

LES OBJETS DE BASE : DEPLOYMENT

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grr
  namespace: test-ns
spec:
  replicas: 3 # Le nombre de réplicas
  selector:
    matchLabels:
      app: grr
  template: # Un template de pod
    metadata:
      labels: # Des labels...
      app: grr
    spec:
      containers:
        image: registry.plmlab.math.cnrs.fr/anf2024/grr:v4.3.5-docker-8
        imagePullPolicy: IfNotPresent
        name: grr
        ports:
          - containerPort: 8080
            name: web
            protocol: TCP
```

- StatefulSet : réseau, stockage et nom d'hôte persistants. Utile pour les applications statefull (et/ou en cluster).
- DaemonSet : un pod par nœud. Utile pour le monitoring, les logs, le stockage...

Ressource durable qui permet d'exposer les pods et la découverte de service.

Adresse IP et nom DNS statiques

```
apiVersion: v1
kind: Service
metadata:
  name: grr
  namespace: test-ns
spec:
  ports:
    - name: web
      port: 8080
      protocol: TCP
      targetPort: 8080
  selector: # Sélection des pods par label
    app: grr
```

- Job, CronJob : gestion des tâches.
- Volume, PersistentVolume, PersistentVolumeClaim : gestion des volumes persistants.
- ConfigMap, Secret : gestion de la configuration et des secrets.
- Ingress : gestion du reverse-proxy.

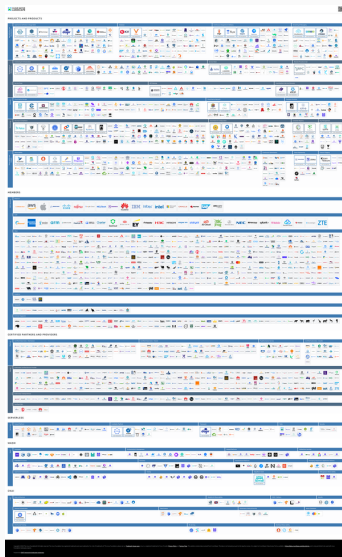
- CRD : extension de l'API Kubernetes, de nouveaux objets
- Custom Controllers
- Operator Pattern : automatisation des processus de déploiements en utilisant CRDs et Custom Controllers
- Exemples : Elasticsearch, PostgreSQL, Prometheus, ...
- Voir <https://operatorhub.io/>

- Gestion de packages (*charts*) pour Kubernetes.
- En fait, un moteur de templating avec la possibilité de publier les charts.
- Devenu le standard pour la distribution d'application.
- Des alternatives existent : kustomize, jsonnet, ksonnet.

- GitOps : gestion de l'infrastructure et des configurations d'applications qui reposent sur l'utilisation de Git.
- Une unique source de vérité pour la formalisation déclarative de l'infrastructure et des applications.
- Fonctionne avec kustomize, Helm, jsonnet, ksonnet.
- Intégration d'outils tiers facile.
- Bootstrap de cluster !

- Projet de la Linux Foundation.
- Lancé en 2015 pour aider à faire progresser les technologies conteneurs.
- 752 membres, dont Google, Red Hat, Huawei, Intel, Cisco, IBM, et VMware.
- Pilotage de Kubernetes depuis 2018.
- 26 projets Graduated, 36 projets Incubating, 124 projets Sandbox.

CNCF : CLOUD NATIVE COMPUTING FOUNDATION



Source: <https://landscape.cncf.io/>

CONCLUSION

- Fiable, robuste, complet, extensible, pas si complexe.
- De nouveaux paradigmes, un écosystème très (très (très)) dynamique...

... mais ne résoud pas tout vos problèmes !



Source: dilbert.com