Université Sorbonne Paris Nord
École d'ingénieur Sup Galilée

Du développement au déploiement d'applications web

Cours Minikube

Réalisé par :

Célian WIRTZ (celian.wirtz@edu.univ-paris13.fr)

3ème année
Année universitaire : 2025/2026

Objectif et contexte

- Le but est de comprendre à quoi sert Kubernetes (K8s) et pourquoi on l'utilise plutôt qu'un simple orchestrateur local comme Docker Compose.
- L'exercice montre aussi comment tester K8s en local avec Minikube, puis déployer une application simple (Nginx), la mettre à l'échelle, l'exposer, et nettoyer les ressources.

Docker Compose vs Kubernetes (idée générale)

- Docker Compose sert surtout à décrire et lancer une application multi-conteneurs sur une seule machine, souvent pour du dev/test local.
- Kubernetes sert à orchestrer des conteneurs sur un cluster (plusieurs machines possibles), avec des mécanismes natifs de haute dispo, réparation automatique et montée/descente en charge.

Concepts Kubernetes à retenir

- **Pod** : plus petite unité déployable, contenant un ou plusieurs conteneurs qui partagent réseau (IP/ports) et stockage du Pod.
- Les Pods sont éphémères : s'ils meurent/si on les remplace, ils reviennent avec une nouvelle IP, donc on ne doit pas dépendre directement de l'IP d'un Pod.
- **Service** : donne un point d'accès stable (IP + DNS) vers un ensemble de Pods, et fait l'équilibrage de charge entre eux.
- Types cités : ClusterIP (interne), NodePort (port exposé sur chaque nœud), LoadBalancer (IP publique via un cloud).
- **Ingress** : joue le rôle de "routeur HTTP/HTTPS" (couche 7) pour exposer des Services vers l'extérieur avec des règles par nom de domaine et/ou chemin, et peut gérer TLS.**Architecture d'un cluster (qui fait quoi)**
- Le cluster est séparé en **Control Plane** (pilotage) et **Worker Nodes** (exécution des Pods).

- Dans le Control Plane :
    - API Server = point d'entrée des commandes/communications.
    - etcd = base qui stocke l'état et la config du cluster.
    - Scheduler = décide sur quel nœud placer les nouveaux Pods.

- Dans un Worker Node :
    - kubelet = agent local qui lance/maintient les Pods demandés.
    - kube-proxy = met en place les règles réseau pour la connectivité Pods/Services.
    - container runtime (Docker/containerd…) = exécute réellement les conteneurs.

Minikube : manip et commandes observées

- Minikube sert à lancer un cluster Kubernetes local (souvent “1 nœud” où control-plane et worker sont réunis), idéal pour apprendre et tester.
- Démarrage : minikube start (dans tes notes, utilisation du driver Docker sur Windows, téléchargement/préparation de Kubernetes, activation d’addons de stockage).
- Vérification versions : minikube version et kubectl version --client, avec un point d’attention sur un possible décalage de version de kubectl par rapport au serveur K8s.
- Vérifier le cluster : kubectl get nodes (le nœud “minikube” apparaît Ready).

Déploiement “hello-nginx” (approche impérative)

- Créer un déploiement : kubectl create deploiement hello-nginx --image=nginx, puis vérifier avec kubectl get pods.
- Mise à l’échelle : kubectl scale deploiement hello-nginx --replicas=4 (création de plusieurs Pods identiques).
- Exposition réseau : kubectl expose deploiement hello-nginx --type=NodePort --port=80 pour créer un Service accessible via un port du nœud.
- Accès simplifié : minikube service hello-nginx --url pour obtenir l’URL locale et tester dans le navigateur (note : avec le driver Docker sur Windows, le terminal doit rester ouvert pour que le tunnel/forward reste actif).
- Nettoyage : suppression du Service puis du déploiement via kubectl delete service ... et kubectl delete deploiement ..., puis vérification qu’il n’y a plus de Pods.

Déploiement via YAML

- On utilise fichier YAML deploiement (nom, nombre de replicas, labels, image nginx, port conteneur), appliqué via kubectl apply -f ....
- L’éphémérité/self-healing est illustrée : si on supprime un Pod à la main (kubectl delete pod ...), il ne reste pas “mort” longtemps car le déploiement recrée un Pod pour revenir à l’état désiré.
- Fin de session : « minikube stop » pour arrêter le cluster local.