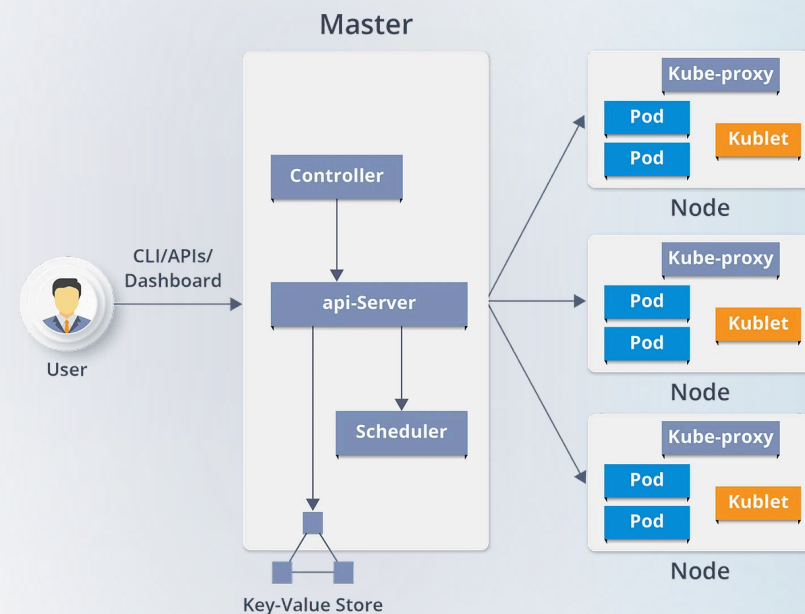


04 - Déploiement et Maintenance d'Applications

Ce chapitre aborde le déploiement d'applications dans Kubernetes et les pratiques essentielles pour les maintenir en production, notamment la mise à l'échelle, les mises à jour continues et les stratégies de surveillance.

 **par Tancrède SUARD**

Code : KUB-A-1 (2024)



Déploiement d'une Application Basique

Définition du Déploiement

Un déploiement dans Kubernetes définit l'état souhaité pour un ensemble de pods.

Gestion des Pods

Kubernetes maintient cet état en gérant le nombre de réplicas et en appliquant les mises à jour des conteneurs.

Avantages

Le déploiement permet de déployer et de maintenir facilement des applications dans un environnement Kubernetes.

Création d'un Déploiement

1

Définir le Déploiement

Créez un objet Deployment qui spécifie le nombre de pods, l'image du conteneur et d'autres configurations.

2

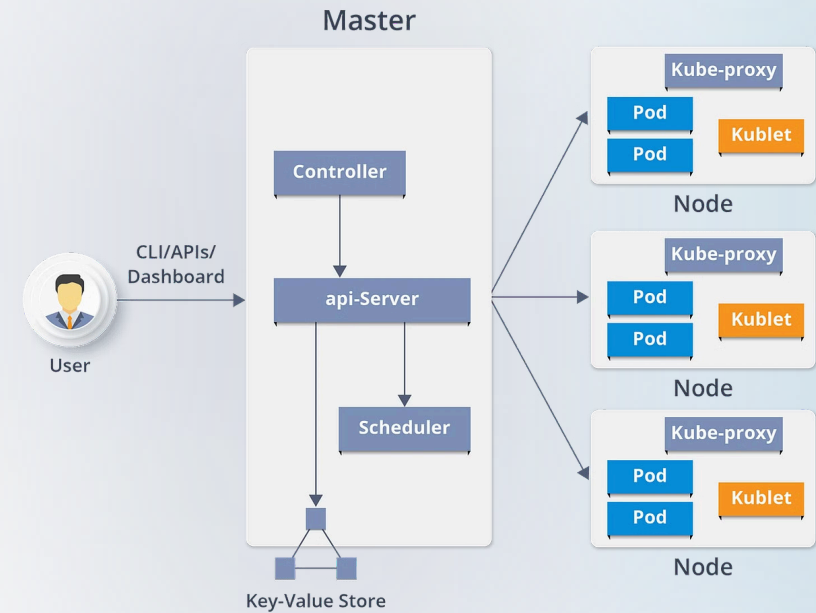
Exemple de Fichier

Utilisez un fichier deployment-example.yaml pour définir les paramètres du déploiement.

3

Appliquer le Déploiement

Exécutez la commande `kubectl apply -f deployment-example.yaml` pour déployer l'application.



Vérification et Gestion du Déploiement

1

Lister les Déploiements

Utilisez la commande **kubectl get deployments** pour afficher la liste de vos déploiements.

2

Détails du Déploiement

Utilisez **kubectl describe deployment my-deployment** pour obtenir les détails d'un déploiement spécifique.

3

Vérifier les Pods

Utilisez **kubectl get pods -l app=my-app** pour voir les pods créés par le déploiement.

Mise à l'Échelle des Applications

Kubernetes permet d'ajuster le nombre de réplicas d'un déploiement pour répondre aux variations de la demande.



Mise à l'Échelle Manuelle

Augmenter

Ajouter des réplicas

Réduire

Diminuer le nombre de réplicas

Commande

```
kubectl scale deployment my-deployment --replicas=5
```

Vous pouvez manuellement ajuster le nombre de réplicas pour un déploiement à l'aide de la commande `kubectl scale`.

Mise à l'Échelle Automatique avec l'HPA

Configuration

Définir les paramètres d'utilisation de la CPU

Ajustement Automatique

Maintenir l'utilisation de la CPU autour de 50%

Limites

Minimum de 2 pods, maximum de 10 pods

L'Horizontal Pod Autoscaler (HPA) ajuste automatiquement le nombre de pods en fonction des besoins en ressources, comme l'utilisation de la CPU. Il permet de maintenir l'utilisation de la CPU autour d'un seuil défini, avec des limites minimum et maximum de pods.

Mises à Jour Continues et Roulement

Kubernetes facilite les mises à jour continues (rolling updates) sans interruption du service.

Mises à Jour avec Rolling Update

1

Déploiement par Défaut

Kubernetes utilise le rolling update pour les déploiements par défaut.

2

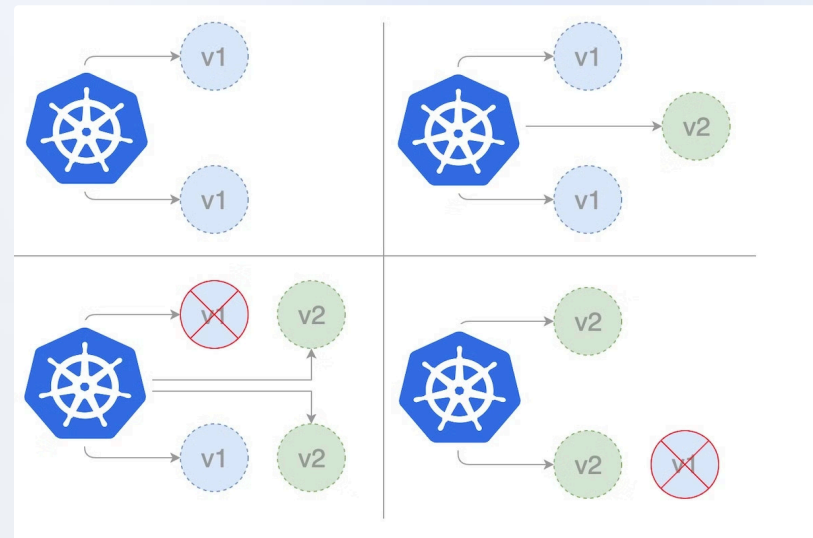
Mise à Jour de l'Image

Mettez à jour l'image d'un déploiement avec la commande **kubectl set image**.

3

Remplacement Progressif

Kubernetes crée de nouveaux pods avec la nouvelle image et arrête progressivement les anciens.



Revert et Rollback

Revenir en Arrière

En cas de problème avec une mise à jour, vous pouvez revenir à une version précédente avec la commande **rollback**.

Historique des Déploiements

Utilisez **rollout history** pour vérifier l'historique des déploiements et gérer les versions.

Surveillance et Logs

Surveiller l'état des applications est essentiel pour détecter rapidement les problèmes. Kubernetes offre des outils pour visualiser les logs et surveiller les ressources.



Consultation des Logs des Pods



Afficher les Logs d'un Pod

Utilisez la commande **kubectl logs <POD_NAME>** pour consulter les logs d'un pod spécifique.

Visualiser les Logs en Temps Réel

Ajoutez l'option **-f** pour suivre les logs en temps réel : **kubectl logs -f <POD_NAME>**.

Identifier les Problèmes

Les logs peuvent aider à identifier les problèmes dans les applications conteneurisées.

Utilisation de Probes pour Vérifier l'État des Applications

1 Liveness Probe

Vérifie si le pod doit être redémarré en cas de défaillance.

2 Readiness Probe

Indique si le pod est prêt à recevoir du trafic.

3 Configuration des Probes

Exemple de configuration de probes dans un déploiement.

Bonnes Pratiques pour le Maintien des Applications



Limites de Ressources

Définir des limites de CPU et de RAM pour éviter l'épuisement des ressources.



Mises à Jour Automatisées

Utiliser des pipelines CI/CD pour déployer automatiquement les nouvelles versions.



Surveillance des Métriques

Intégrer des outils de monitoring pour détecter les anomalies.



Alertes Configurées

Mettre en place des alertes pour être notifié en cas de défaillances.

