06 - Kubernetes Avancé et Bonnes Pratiques

Ce chapitre explore les fonctionnalités avancées de Kubernetes pour optimiser les performances, garantir la haute disponibilité et faciliter la maintenance du cluster. Il présente également les meilleures pratiques pour gérer un environnement Kubernetes en production.

 **par Tancrède SUARD**

Code : KUB-A-1 (2024)

Gestion de la Scalabilité et de la Haute Disponibilité

Scalabilité

Kubernetes permet de faire évoluer les ressources du cluster pour répondre aux besoins variables des applications.

Haute Disponibilité

La haute disponibilité assure la résilience des applications en cas de pannes ou de surcharge.

Résilience

La combinaison de la scalabilité et de la haute disponibilité garantit la résilience des applications Kubernetes.

Haute Disponibilité du Plan de Contrôle

ETCD Redondant

Déployer ETCD sur plusieurs nœuds pour assurer la persistance des données.

Load Balancer pour les API Servers

Utiliser un load balancer pour répartir les requêtes entre les API Servers.

Nœuds Maîtres Redondants

Configurer plusieurs nœuds maîtres (API Servers, Scheduler, Controller Manager) pour garantir la haute disponibilité du cluster.

Autoscaling du Cluster

Horizontal Pod Autoscaler (HPA)

Ajuste automatiquement le nombre de pods pour un déploiement spécifique.

Cluster Autoscaler

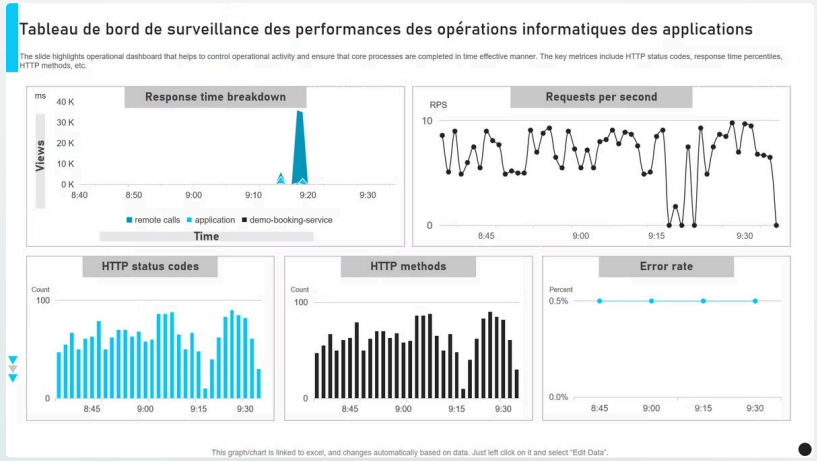
Ajuste le nombre de nœuds dans le cluster en fonction des besoins en ressources.

Configuration

Utilisez les configurations spécifiques de votre fournisseur de cloud ou installez l'outil Cluster Autoscaler pour les environnements on-premise.

Surveillance et Gestion des Logs

La surveillance des métriques et la gestion des logs sont essentielles pour la gestion proactive des applications et pour le diagnostic des problèmes en production.



Intégration de Prometheus et Grafana

1. **Déployer Prometheus**
   Avec Helm ou fichiers YAML
2. **Configurer Grafana**
   Visualiser les métriques
3. **Créer des Alertes**
   Être notifié en cas de dépassement de seuils

Prometheus collecte les métriques sur les ressources et les performances des applications. Grafana permet de visualiser ces métriques et de créer des tableaux de bord et des alertes en temps réel.

Gestion des Logs avec Fluentd ou Elastic Stack

Fluentd

Collecte et centralise les logs des applications et de Kubernetes.

Elastic Stack

Stocke, analyse et visualise les logs pour faciliter le débogage.

Bonnes Pratiques de Déploiement et de Gestion de Kubernetes

Pour garantir la stabilité, la sécurité et la maintenabilité du cluster, voici quelques bonnes pratiques à suivre.

Organiser et Versionner les Manifests

Helm

Créez des packages d'applications, appelés charts, pour gérer les versions et déployer les applications avec leurs dépendances.

Kustomize

Intégrez les configurations spécifiques à différents environnements (production, staging, développement).

Gestion de Configuration

Utilisez des outils comme Helm ou Kustomize pour organiser les manifests (fichiers YAML) et faciliter le déploiement.

Gérer les Ressources avec les Limites et les Quotas

Requests

Ressources minimales requises pour démarrer un pod.

Limites

Limites maximales de consommation de ressources d'un pod.

Quotas

Empêchent les pods de monopoliser les ressources du cluster.

Sécuriser les Images des Conteneurs

Scanner les Images

Utilisez des outils comme Clair ou Trivy pour détecter les vulnérabilités dans les images.

Utiliser des Images de Confiance

Optez pour des images vérifiées dans des registres sécurisés.

Restreindre les Permissions

Évitez d'utiliser des conteneurs qui s'exécutent avec des privilèges élevés.

Automatiser les Déploiements avec CI/CD

1

Configurer les Pipelines

Utiliser des outils comme **Jenkins** ou **GitLab CI** pour automatiser la construction, les tests et le déploiement des conteneurs.

2

Adopter GitOps

Argo CD fournit un système de déploiement GitOps, où les configurations du cluster sont gérées directement dans un dépôt Git.

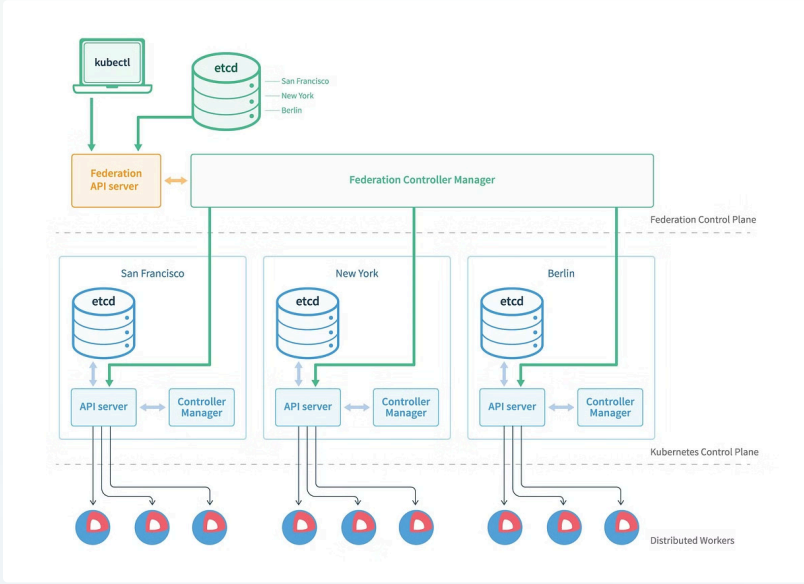
3

Réduire les Erreurs

L'automatisation des déploiements permet de réduire les erreurs manuelles et de garantir la cohérence entre les versions.

Optimisation des Performances

Pour maintenir la performance et la propreté du cluster, il est important de régulièrement optimiser les ressources et de nettoyer les objets inutilisés.



Optimiser l'Utilisation des Ressources

Analyse de l'Utilisation

Utilisez **kubectl top** pour analyser l'utilisation des ressources par les pods.

Ajuster les Limites

Ajustez les limites de CPU et de mémoire pour chaque pod en fonction de ses besoins réels.

Supprimer les Objets Inutilisés

Commande kubectl delete

Utilisez cette commande pour supprimer les ressources spécifiques qui ne sont plus nécessaires.

Garbage Collection

Kubernetes gère automatiquement la suppression des objets orphelins, mais il est conseillé d'effectuer des nettoyages réguliers.

Éviter l'Encombrement

Il est important de supprimer régulièrement les ressources non utilisées pour éviter l'encombrement du cluster.