

03 - Premiers pas avec les Objets Kubernetes

Explorons les objets de base de Kubernetes qui permettent de déployer, exposer et gérer des applications dans un cluster. Ces concepts sont essentiels pour comprendre comment Kubernetes orchestre les conteneurs.



par Tancrede SUARD

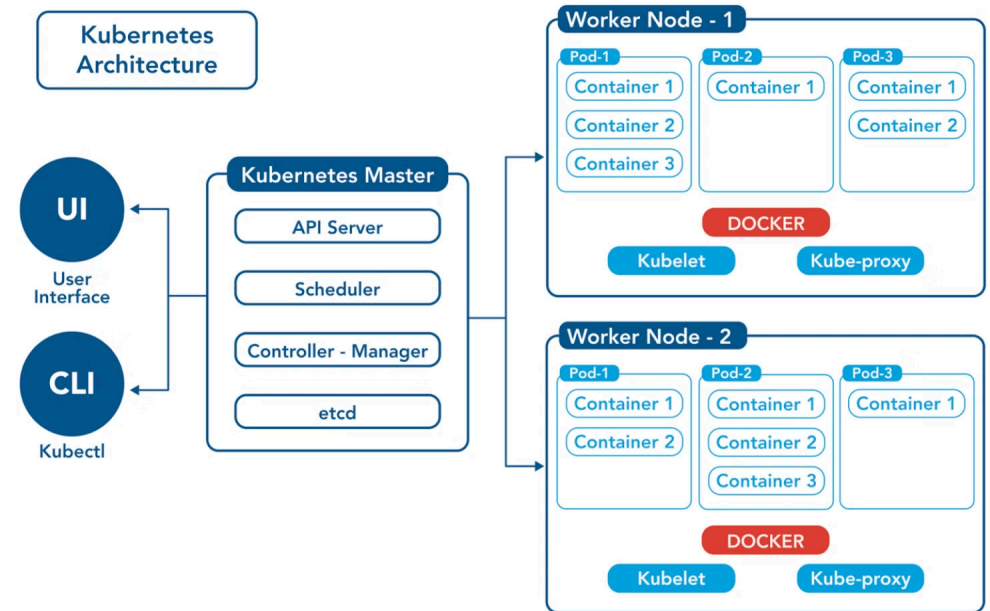
Code : KUB-A-1 (2024)

Concepts Fondamentaux des Objets Kubernetes

Pod	Service	Volume	Namespace
L'unité de déploiement de base, contenant un ou plusieurs conteneurs partageant les mêmes ressources.	Expose les applications pour permettre la communication entre les pods et avec l'extérieur.	Fournit un stockage persistant pour les applications au sein des pods.	Segmente le cluster pour isoler des groupes de ressources et faciliter la gestion d'environnements.

Les Pods : L'Unité Fondamentale

Un **pod** est la plus petite unité de déploiement dans Kubernetes. Chaque pod contient généralement un seul conteneur, mais plusieurs pods peuvent être créés pour une même application, chacun étant géré individuellement par Kubernetes.



Création d'un Pod

1

Fichier de configuration YAML

Créez un fichier YAML qui spécifie les détails du conteneur et ses configurations.

2

Définir les paramètres

Incluez les informations telles que le nom du pod, les étiquettes et les spécifications des conteneurs.

3

Déployer le pod

Exécutez la commande **kubectl apply -f pod-example.yaml** pour créer le pod.

Vérification et Surveillance des Pods

1

Lister les Pods

Utilisez la commande **kubectl get pods** pour afficher la liste des pods en cours d'exécution.

2

Obtenir les Détails d'un Pod

Utilisez **kubectl describe pod my-pod** pour obtenir des informations détaillées sur un pod spécifique.

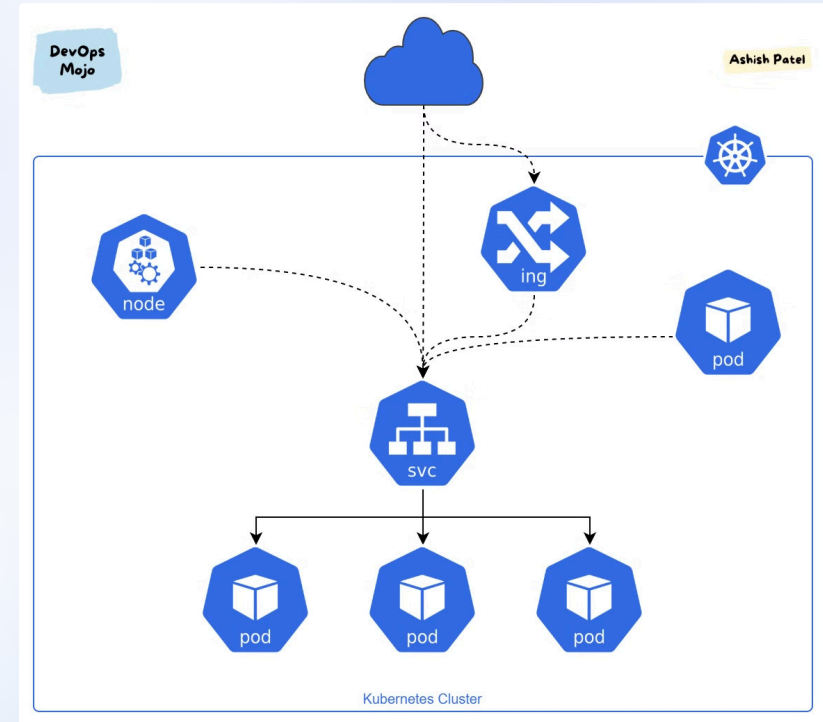
3

Visualiser les Logs

Utilisez **kubectl logs my-pod** pour afficher les journaux d'un conteneur dans un pod.

Services : Exposer les Applications

Les **services** permettent d'exposer les pods pour permettre aux applications de communiquer entre elles et avec l'extérieur.



Types de Services

ClusterIP

Accessible uniquement depuis le cluster, pour une communication interne.

NodePort

Expose le service sur un port spécifique de chaque nœud du cluster.

LoadBalancer

Expose le service via un équilibreur de charge (disponible principalement dans les environnements cloud).

Création d'un Service pour un Pod

1

Définir le Service

Créer un fichier YAML avec les paramètres du service

2

Appliquer le Service

Exécuter la commande `kubectl` pour déployer le service

3

Exposer le Pod

Le service permet d'exposer le pod créé précédemment

Vérification de l'Accessibilité du Service



Lister les Services

Utilisez la commande **kubectl get services** pour afficher la liste des services disponibles.



Tester l'Accès

Testez l'accès au service en utilisant l'IP du cluster et le port exposé.



Vérifier le Fonctionnement

Assurez-vous que le service fonctionne correctement en testant son accessibilité.

Volumes et Persistance des Données

Volumes Kubernetes

Kubernetes permet d'associer des volumes aux pods pour conserver les données même si un pod est supprimé ou recréé.

Persistance des Données

Les volumes Kubernetes assurent la persistance des données au-delà de la durée de vie des pods.

Cycle de Vie des Pods

Les volumes permettent de préserver les données lorsque les pods sont supprimés et recréés.

Types de Volumes dans Kubernetes



emptyDir

Volume temporaire pour stocker des données pendant la durée de vie du pod.

hostPath

Monte un répertoire du nœud dans le pod.

PersistentVolume (PV) et PersistentVolumeClaim (PVC)

Volumes persistants indépendants du cycle de vie des pods.

Création d'un Volume PersistentVolume (PV) et PersistentVolumeClaim (PVC)

1 PersistentVolume

Définition des capacités et des modes d'accès

2 PersistentVolumeClaim

Demande de stockage avec des critères spécifiques

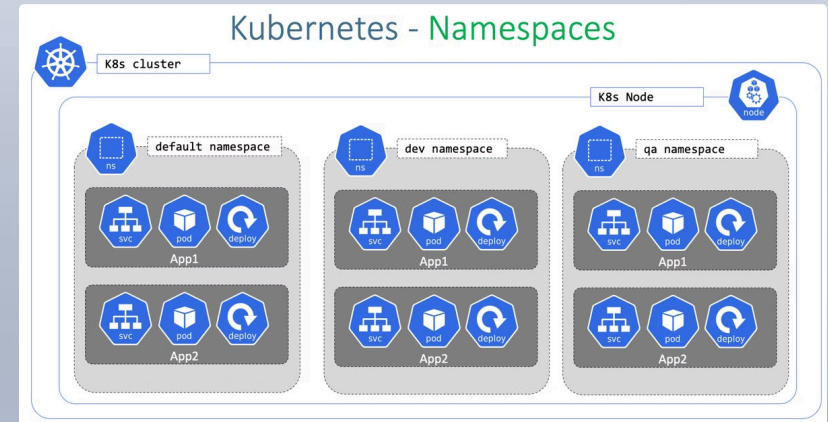
3 Utilisation dans un pod

Le PVC est prêt à être utilisé pour assurer la persistance des données

Le PersistentVolume (PV) définit les capacités et les modes d'accès du stockage, tandis que le PersistentVolumeClaim (PVC) est une demande de stockage avec des critères spécifiques. Une fois le PVC créé, il peut être utilisé dans un pod pour assurer la persistance des données.

Namespaces : Gestion des Environnements

Les **namespaces** permettent de segmenter les ressources du cluster pour isoler des groupes de pods et d'objets en fonction des besoins (environnement de production, de développement, etc.).



Création d'un Namespace

1

Commande kubectl

Exécutez la commande **kubectl create namespace dev-environment** pour créer un nouveau namespace nommé "dev-environment".

2

Fichier YAML

Définissez le namespace dans un fichier YAML et appliquez-le avec **kubectl apply -f namespace.yaml**.

3

Utilisation du Namespace

Une fois créé, vous pouvez utiliser ce namespace pour déployer vos ressources Kubernetes.

Utilisation des Namespaces

Commandes avec Namespace

Pour gérer des ressources dans un namespace spécifique, ajoutez le paramètre `-n` dans vos commandes.

Exemple

```
kubectl get pods -n dev-environment
```