

Nom : Sahari  
Prénom : Mehdi

## Développement et services cloud

Quelle est la différence entre un service ClusterIP et NodePort sous Kubernetes ?

La différence entre un service ClusterIP et NodePort sous Kubernetes est que ClusterIP expose le service uniquement à l'intérieur du cluster, le rendant accessible seulement aux autres services et pods au sein du cluster via une adresse IP interne alors que NodePort expose le service sur chaque nœud du cluster à un port statique, le rendant accessible de l'extérieur du cluster via l'IP du nœud et le port spécifique.

Quel type de scaling je choisirai pour une BDD ? Et pourquoi ?

Le type de scaling que je choisirai pour une BDD serait un scaling horizontal.

Car il permet une :

- **Meilleure gestion de la charge** : Permet de répartir la charge de travail sur plusieurs instances, ce qui améliore la performance et la tolérance aux pannes.
- **Scalabilité** : Offre une capacité de croissance presque illimitée en ajoutant simplement plus d'instances.
- **Disponibilité et Redondance** : En cas de panne d'une instance, d'autres instances peuvent prendre le relais, assurant une meilleure disponibilité.

Pour quelles raisons choisir un cloud privé plutôt que publique ?

Les raisons pour lesquelles je choisirai un cloud privé plutôt que publique sont :

- **La sécurité** : Meilleur contrôle sur les données et plus facile de répondre aux exigences réglementaires.
- **La flexibilité de personnalisation** : Plus de flexibilité pour configurer le matériel et les logiciels selon des besoins spécifiques.
- **La performance** : Ressources dédiées, réduisant la latence et les fluctuations de performance.
- **L'isolation** : Pas de partage de ressources avec d'autres utilisateurs.
- **La connaissance des coûts** : Moins de coûts variables, ce qui est avantageux pour des charges de travail stables et prévisibles.

Dans quel cas prendre une instance AWS réservée ? Et quel est l'intérêt ?

Je prendrais une instance AWS réservée dans le cas où j'ai des tâches régulières et constantes qui nécessitent toujours des ressources disponibles avec des besoins de capacité bien définis et à long terme.

Les intérêts ici seraient :

- **Économiques** : Permet de réaliser des économies significatives par rapport aux instances à la demande, avec des réductions pouvant aller jusqu'à 70%. Les coûts sont également fixes et prévisibles sur une période d'un ou trois ans.
- **Garantie de capacité** : AWS réserve les ressources, assurant leur disponibilité constante et ce même pendant les périodes de forte demande. Donc pas de risque de pénurie ou de perte d'accès aux services en production.

Vous réaliserez deux micro services (deux API) déployés dans Kubernetes. Le premier micro service est exposé en dehors du cluster, c'est une API que vous appellerez et qui appellera une seconde API via le réseau du cluster pour vous remonter la réponse.

Le rendu attendu sont le dépôt git incluant : Les deux API et les fichiers de configuration YML Kubernetes.

Tel que :

