

Brief 6 - part 1 : Kubernetes

Contexte du projet

Vous allez déployer l'application **Azure Voting App** et sa base de données **Redis** sur le cluster **Kubernetes** de **Azure** : **AKS**.

Ce brief est individuel. Le rendu sera aussi individuel. Vous êtes bien sûr invités à vous entraider.



RIEN NE DEVRA ÊTRE DÛ AU HASARD.

COPIER-COLLER INTERDIT SI PAS COMPRIS.

CHACQUE ERREUR DEVRA ÊTRE LUE ENTIÈREMENT AVANT D'ÊTRE SOUMISE AU FORMATEUR.



Modalités pédagogiques

Chapitre 1 : Déployer un cluster AKS

1. Créer un cluster **AKS** avec 2 **nodes**

Chapitre 2 : Déployer un container Redis

1. Créer un **Deployment** avec une image **Redis**
2. Créer un **Service** de type **LoadBalancer** pour exposer le **Deployment** sur le port 6379
3. Installer un client **Redis** sur votre poste de travail
4. Tester la connexion au container **Redis** déployé

Chapitre 3 : Déployer un container Voting App

1. Créer un **Deployment** avec une image de l'application **Azure Voting App** qui dépend de **Redis**
2. Configurer la variable d'environnement **STRESS_SECS** à **2** pour le container **Voting App**
3. Créer un **Service** de type **LoadBalancer** pour exposer le container **Azure Voting App**
4. Modifier le type du **Service** de **Redis** pour utiliser **ClusterIP**
(<https://learn.microsoft.com/fr-fr/azure/aks/concepts-network> (<https://learn.microsoft.com/fr-fr/azure/aks/concepts-network>))

fr/azure/aks/concepts-network))

5. Tester le bon fonctionnement de l'application depuis votre navigateur web

Chapitre 4 : Un mot de passe pour Redis

1. Configurer le container **Redis** pour authentifier les clients avec un mot de passe (utiliser `--requirepass` pour démarrer `redis-server`)
2. Le mot de passe à utiliser devra être sécurisé dans un **Kubernetes Secret**
3. Configurer le container **Voting App** pour utiliser ce mot de passe avec la variable d'environnement **REDIS_PWD**

Chapitre 5 : Configurer un stockage persistant pour Redis

1. Supprimer puis recréer le container **Redis**
2. Constater que le compte de votes est remis à 0 sans avoir à cliquer sur *Reset*
3. Déduire que le stockage est volatile par défaut
4. Créer un **PersistentVolumeClaim** pour le container **Redis** avec une **StorageClass** adéquate pour permettre le scale out de Redis dans un futur brief
5. Refaire le test des points 1. et 2. et en conclure que le stockage est maintenant permanent

Critères de performance

- Le code est lisible (clair et facilement compréhensible)
- Les configurations s'appliquent correctement et déploient automatiquement l'application
- La méthode Scrum a été suivie
- Rien n'a été écrit "par hasard"

Modalités d'évaluation

Restitution individuelle.

Relecture commentée de vos livrables par les formateurs.

Livrables

- Lien vers le dépôt des sources
- DAT (Document d'Architecture Technique) de l'infrastructure déployée
- un executive summary de votre travail (technique et organisationnel)
- dans votre executive summary, vous présenterez le fonctionnement de Kubernetes