

# Brief 8 - Complete CI/CD

---

## Contexte du projet

---

Vous allez déployer un pipeline d'intégration et de déploiement continu pour l'application **Azure Voting App** et sa base de données **Redis**.

À chaque mise à jour du code source de l'application, le pipeline devra automatiquement builder une image docker et la déployer en passant par une phase de test et un processus de canary release.

Pour cela, vous aller forker le dépôt `azure-vote` (<https://github.com/simplon-alfred/azure-voting-app-redis>) et compléter votre pipeline CI/CD du brief 7.

Vous pouvez utiliser le script `auto-maj.sh` (<http://auto-maj.sh>) pour simuler un changement de l'application (mis à jour du numéro de version).

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

Ce brief durera 1 semaine.

## Modalités pédagogiques

---

### 1) Création d'un plan de test du déploiement

Il s'agit ici de tester le bon fonctionnement du déploiement, pas du logiciel.

Vous allez créer 2 environnements : QAL et PROD complètement séparé dans des namespaces k8s différents. L'idée est de déployer d'abord sur QAL et si les tests automatiques passent, déployer alors sur PROD, le tout automatiquement dans la même pipeline.

### 2) Écriture des tests automatiques

### 3) Écriture de la pipeline de déploiement

### 4) Déployer en canary release

### 5) Déploiement en GA

## Critères de performance

---

- Le code est lisible (clair et facilement compréhensible)
- Les configurations s'appliquent correctement et déploient automatiquement l'application
- Rien n'a été écrit "par hasard"
- Les bonnes pratiques de MCO et MCS ont été respectée

## Modalités d'évaluation

---

Restitution collective et commentée.

## Livrables

---

- Lien vers le dépôt des sources
- Procédure d'update