

ESGI - 3AL - 2024/2025 - Outils de développement et conteneurisation avancée - Projet : Papermerge

Objectif du projet

L'objectif de ce projet est de conteneuriser une application distribuée en utilisant Kubernetes.

Vous devrez conteneuriser l'application Papermerge.

[Papermerge](#) est une application de type Document Management Software (DMS), qui permet de stocker des documents d'entreprise, qu'ils soient manuscrits ou informatisés.

Exercice

Vous devrez créer une configuration Kubernetes permettant d'exécuter l'application sur un cluster Minikube local.

Cette application a besoin de 6 composants pour s'exécuter correctement.

La description de ces 6 composants sera détaillée dans les sous-parties ci-dessous.

Remarques

- Vous pouvez découper la configuration Kubernetes du projet en plusieurs fichiers
- Vous devrez fournir un dépôt git contenant les fichiers de configuration Kubernetes réalisés. Chaque membre du groupe devra avoir contribué équitablement au projet.
- Lors de la soutenance, vous devez arriver avec votre application déployée sur un cluster Minikube local et qui fonctionne correctement. Votre intervenant testera le bon fonctionnement de l'application.
- Vous aurez des questions sur ce que vous avez produit durant la soutenance.

Caractéristiques de l'application

Base de données

Paperless a besoin d'une base de données PostgreSQL en version 17 pour fonctionner.

Le conteneur aura les caractéristiques suivantes :

- Image : `postgres:17-alpine`
- Port : `5432`
- Stockage de données : `/var/lib/postgresql/data`

Le conteneur doit pouvoir être mis à l'échelle horizontalement, et joignable par les autres conteneurs dans le cluster.

Variables d'environnement à définir

```
POSTGRES_DB: papermerge
POSTGRES_PASSWORD: pmg-secret
POSTGRES_USER: papermerge
```

Stockage clé-valeur (Redis)

Paperless a besoin d'une base de données clé-valeur Redis en version 7 pour fonctionner.

Le conteneur aura les caractéristiques suivantes :

- Image : `redis:7-alpine`
- Port : `6379`
- Stockage de données : *Aucun*

Le conteneur doit pouvoir être mis à l'échelle horizontalement, et joignable par les autres conteneurs dans le cluster.

Variables d'environnement à définir

Aucune

Stockage d'objet (Minio)

Paperless utilise solution de stockage objet (Minio) pour la sauvegarde des documents.

Le conteneur aura les caractéristiques suivantes :

- Image : `minio/minio:latest`
- Ports : `9000` (API), `9001` (Console)
- Stockage de données : `/data`
- Arguments de la commande à exécuter : `server --console-address :9001 /data`

Le conteneur doit pouvoir être mis à l'échelle horizontalement, et joignable par les autres conteneurs dans le cluster **ET** à l'extérieur du cluster.

Variables d'environnement à définir

```
MINIO_ROOT_USER: 2e8817266261610810596a50b8f4a8c5
MINIO_ROOT_PASSWORD: a503071debfa3ac0c1de1cd79f
```

Worker S3 Papermerge

Paperless utilise un worker de synchronisation S3 pour certaines fonctionnalités.

Le conteneur aura les caractéristiques suivantes :

- Image : `papermerge/s3worker:0.4.3`
- Ports : *aucun*
- Stockage de données : `/var/media/pmg`

Le conteneur doit pouvoir être mis à l'échelle horizontalement.

Variables d'environnement à définir

```
AWS_ACCESS_KEY_ID: 2e8817266261610810596a50b8f4a8c5
AWS_REGION_NAME: us-east-1
AWS_S3_ENDPOINT: http://paperless-minio:9000
AWS_SECRET_ACCESS_KEY: a503071debfa3ac0c1de1cd79f
PAPERMERGE__DATABASE__URL: postgresql://papermerge:pmg-secret@papermerge-
database:5432/papermerge
PAPERMERGE__MAIN__MEDIA_ROOT: /var/media/pmg
PAPERMERGE__REDIS__URL: redis://papermerge-redis:6379/0
PAPERMERGE__S3__BUCKET_NAME: papermerge
S3_WORKER_ARGS: "-Q s3,s3preview -c 2"
```

Worker OCR Papermerge

Paperless utilise un worker OCR (Optical Character Recognition) pour certaines fonctionnalités.

Le conteneur aura les caractéristiques suivantes :

- Image : `papermerge/ocrworker:0.3.1`
- Ports : *aucun*
- Stockage de données : *aucun*

Le conteneur doit pouvoir être mis à l'échelle horizontalement.

Variables d'environnement à définir

```
AWS_ACCESS_KEY_ID: 2e8817266261610810596a50b8f4a8c5
AWS_REGION_NAME: us-east-1
AWS_S3_ENDPOINT: http://paperless-minio:9000
AWS_SECRET_ACCESS_KEY: a503071debfa3ac0c1de1cd79f
PAPERMERGE__DATABASE__URL: postgresql://papermerge:pmg-secret@papermerge-
database:5432/papermerge
PAPERMERGE__REDIS__URL: redis://papermerge-redis:6379/0
PAPERMERGE__S3__BUCKET_NAME: papermerge
OCR_WORKER_ARGS: "-Q ocr -c 2"
```

Application Papermerge

Papermerge fonctionne via une application web.

Le conteneur aura les caractéristiques suivantes :

- Image : `papermerge/papermerge:3.3.1`
- Port : `80`
- Stockage de données : `/var/media/pmg`

Le conteneur doit pouvoir être mis à l'échelle horizontalement, et joignable par les autres conteneurs dans le cluster **ET** depuis l'extérieur du conteneur.

Variables d'environnement à définir

```
PAPERMERGE__AUTH__USERNAME: admin
PAPERMERGE__AUTH__PASSWORD: secret
PAPERMERGE__DATABASE__URL: postgresql://papermerge:pmg-secret@papermerge-
database:5432/papermerge
PAPERMERGE__MAIN__MEDIA_ROOT: /var/media/pmg
PAPERMERGE__OCR__LANG_CODES: "eng, fra"
PAPERMERGE__OCR__DEFAULT_LANG_CODE: "eng"
PAPERMERGE__REDIS__URL: redis://papermerge-redis:6379/0
PAPERMERGE__SECURITY__SECRET_KEY:
8aa9127e904324e1d93c2b68483a327c3b5be964dd161e3e4df8dbca0443ec21
```

Remarque finale

Si vous n'arrivez pas à faire fonctionner les workers (OCR et/ou S3), ce n'est pas grave. Par contre, il est nécessaire de produire une configuration qui fonctionnerait au plus proche possible de la réalité.

Bon courage !