

# Projet – Leed

## Objectif

L'objectif de ce projet est de réaliser un déploiement complet de l'application Leed.

Au travers de ce projet, il est attendu de vous que vous démontreriez ce que vous avez appris sur `docker`, `docker-compose` et le *packaging* d'applications. Il est attendu de vous que vous mettiez en œuvre les bonnes pratiques du domaine.

## Évaluation & Rendus

Ce projet est à réaliser en monôme. Ce projet est noté. Ce projet est prévu pour deux séances de 2 h.

La suite du sujet fournit un cahier des charges fonctionnel qu'il vous est demandé de respecter. Le non-respect d'un ou plusieurs items de la section 1.1 sera considéré comme un rendu hors-sujet (note maximum de 6/20).

### Rendus

1. dépôt `git` contenant le *packaging*
2. compte rendu de projet

---

## 1 Cahier des charges fonctionnel

L'application Leed « est un agrégateur RSS/ATOM minimaliste qui permet la consultation de flux RSS de manière rapide et non intrusive ». Le dépôt `git` de l'application est publié sur GitHub <https://github.com/LeedRSS/Leed>.

Il vous est demandé de réaliser un *packaging* de cette application en respectant le cahier des charges défini ci-après.

### 1.1 Éléments minimaux

Tous les éléments présents dans cette section doivent être présents. L'oubli/manque d'un des éléments minimaux est un hors-sujet.

**Organisation de l'infrastructure** L'application Leed est une application multi-tiers, aussi votre infrastructure sera découpée en au moins deux services (obligatoire) et potentiellement un service optionnel.

**Maîtrise du *packaging*** Au moins un des service déployés reposera sur une image sur mesure (`Dockerfile`).

**Accès à l'application** L'application sera accessible via un port exposé. **Ce port doit pouvoir être changé !**

**Persistance des données** Les données applicatives seront persistées à l'aide d'un ou plusieurs volumes.

**Configuration centralisée** Tous les éléments de configuration seront présents dans le fichier `.env`.

## 1.2 Éléments complémentaires

**Configuration comportements** Les différents comportements de l'application seront configurables au travers de variables d'environnement. Au moins un comportement doit être configurable pour valider cet élément.

**Séparation des modes d'exécution** Le déploiement en mode production ou développement de l'application seront distingués. Par exemple, en mode développement l'accès au code source peut être facilité et des outils comme `xdebug` peuvent être ajoutés.

**README** Un fichier `README` doit expliquer la configuration minimale nécessaire pour déployer l'application, ainsi que les modifications possibles.

## 2 Consignes rendus

### 2.1 Dépôt git

Il vous a été assigné un groupe GitLab dont vous êtes le propriétaire. L'URL de votre groupe a la structure suivante :

`https://gricad-gitlab.univ-grenoble-alpes.fr/iut2-info-stud/2023-s5/r5-adv-virt/grp-{xx}/{login}`

Remplacez `{xx}` par `a1`, `a2`, `a3`, ou `b1` en fonction de votre groupe. Remplacez `{login}` par votre login.

Vous définirez votre infrastructure dans un (ou plusieurs) dépôt(s) `git` dans votre groupe GitLab.

### 2.2 Compte Rendu

Il vous est demandé de rédiger un compte rendu de projet. Attention, ce compte rendu est un document différent du `README` mentionné dans le cahier des charges.

Dans ce compte rendu, il est attendu que vous documentiez les choix que vous avez fait pour *packager* l'application. En particulier, vous commenterez votre réalisation au regard des bonnes pratiques du domaine. Pour cela, vous pouvez vous aider du manifeste *The Twelve-Factor App* <https://12factor.net/>.

Vous mentionnerez les sources et outils que vous avez utilisé pour votre travail (documentation, billets de blog, FAQ, forums, IA générative, ...). Vous utiliserez les guillemets « » pour indiquer les éléments que vous reprenez dans le corps de texte.