

Log680 - Laboratoire 2

5 juillet 2024

Robyn Girardeau
Nicolas Patenaude

Introduction

Ce deuxième laboratoire consiste en plusieurs parties:

1. Planification du travail
 - Ajouter les tâches aux Kanban
 - Attribuer/Assigner les tâches à Nicolas et Robyn
2. Fork de OxygenCS
 - Faire une fork de OxygenCS, créer les variables d'environnement nécessaires
 -

Répartition des tâches

La répartition des tâche s'est beaucoup mieux déroulée que lors du premier laboratoire, malgré les plusieurs contretemps que nous avons rencontrés. En effet, un des membres de notre équipe, Hugo, nous à quitté. Nous avons donc du se répartir sa charge de travail entre les deux membres restants: Robyn et Nicolas.

Nous avons eu plus de facilité à accomplir les objectifs mentionnés plus tôt car une part de la logique à déjà été mise en place lors du premier laboratoire :

- La structure du API
- La base de données
- Le ORM (Object relationnal mapping)
- Familiarité avec python et les librairies utilisés

Nous nous sommes réparties les tâches beaucoup plus équitablement pour ce laboratoire que pour le premier

- Robyn a mis en place le pipeline de CI/CD et le git hook pre commit avec le linting et l'exécution des tests unitaires. Il à aussi créé la fork de oxygens sur son github.
- Nicolas a fait l'enregistrement dans la base de données des températures et des actions du HVAC dans oxygens. Il a aussi implémenté les métriques de CI/CD cherchés de Docker et de DockerHub.

Arborescence du projet

```

├── .github/
│   ├── workflows
│   │   ├── deploy.yaml
│   │   └── pr.yaml
│   └──
├── .vscode/
│   └── launch.json
├── doc/
│   ├── README.md
│   └── rapport.md
├── src/
│   ├── main.py
│   ├── config.py
│   ├── init.py
│   ├── alchemy/
│   │   ├── alch.py
│   │   ├── alchmodels.py
│   │   └── crud.py
│   ├── models/
│   │   ├── HVACAction.py
│   │   └── Temperature.py
│   └──
├── test/
│   ├── db_test.py
│   ├── hvac_test.py
│   ├── test.py
│   └── init.py
├── .dockerignore
├── .env
├── .gitignore
├── .pre-commit-config.yaml
├── .pylintrc
├── compose.yaml
├── Dockerfile
├── LICENSE
├── Pipfile
├── README.Docker.md
└── README.md

```

Tout d'abord, le fichier `doc\README.md` dans le dossier `doc` est important à lire pour l'installation et l'exécution du projet.

Le fichier `.env` est à créer soit-même, il configure toutes les variables d'environnement et les secrets. Celui-ci doit rester en local uniquement, il est noté dans le `.gitignore`. Il contient notamment le dépôt et le projet à examiner. Plus de détails sur les variables à inclure dans le fichier `.env` se trouvent dans `doc\README.md`

Le dossier `tests/` est explicite, contenant des tests unitaires et d'intégration.

Le dossier `src/` contient quelques fichiers communs:

- `main.py` se trouve dans le dossier `src`, il démarre l'utilitaire OxygenCS.
- `config.py` est responsable de charger les secrets, soit à partir du `.env` (local), soit à partir des

variables d'environnement (Github Actions)

Le dossier `src/alchemy/` contient la gestion de la base de données:

- `alch.py` est responsable de créer le client de base de données
- `alchmodels.py` est responsable des modèles de base de données et de créer les tables automatiquement
- `crud.py` est responsable d'exécuter des requêtes sur la base de données

Le dossier `src/models/` contient les modèles dits Pydantic. Les modèles Pydantic permettent d'être (dé)sérialisés et validés en json au contraire des modèles de base de données. Les modèles de base de données sont automatiquement convertis en modèles Pydantic lors de leur sérialisation à la réponse de requêtes API. Ce projet contient deux modèles:

- HVACAction: Enregistre les actions de l'unité HVAC en fonction de la température et de la configuration de `T_MIN` et `T_MAX`.
- Temperature: Enregistre la température de la pièce en degrés Celcius à un interval prédéfini.

Description et les justifications des étapes d'implémentation de votre CI/CD

Le pipeline CI sur pull request applique le linting, le formattage et les tests d'intégration.

Le pipeline CD sur push sur la branche main construit et déploie une image docker du logiciel sur dockerhub.

Nous utilisons le Dockerfile pour construire cette image.

Par exemple si on veut le construire et rouler manuellement:

```
docker build -t oxygencs .
docker run --env-file .env oxygencs
```

L'image a tout de même besoin de rouler avec le fichier de variables d'environnement `.env` puisqu'il ne peut pas être encapsulé sur github et dockerhub.

Choix de métriques CI

Les métriques CI concernent 3 facettes de notre intégration continues:

Les tags DockerHub

Une image contient un nom et un tag facultatif. Ce tag est un identifiant de manifeste personnalisé et lisible qui correspond généralement à la version spécifique de nos images. Nous allons chercher les informations suivantes sur les tags:

- `tag_id` : identifiant du tag
- `name`: le nom du tag
- `last_updater_username`: le nom d'utilisateur de la personne qui à créé le tag
- `last_updated`: Le moment ou ce tag à été créé
- `status` : l'état du tag

Les conteneurs Dockers

Les informations suivantes sur les conteneurs dockers:

- `container_id`: L'identifiant du conteneur
- `name`: Le nom du conteneur
- `status`: L'état du conteneur
- `build_start`: Le temps du début du build du conteneur
- `build_finish`: Le temps de fin du build du conteneur
- `timestamp`: Le moment où ces informations ont été enregistrés

Les images Docker

Les informations suivantes sur les images dockers:

- `image_id`: identifiant de l'image
- `tag`: Le tag associé à l'image (La version de l'image)
- `os`: Le système d'exploitation qui exécute l'image
- `created`: Le moment de création de l'image
- `project`: Le projet dans lequel l'image se trouve
- `docker_version`: La version de docker qui build et exécute l'image

Conclusion

Bien que nous ayons appris des erreurs que nous avons commises lors du premier laboratoire, ce deuxième laboratoire ne serait probablement pas aussi bien déroulé si nous n'avions pas eu une extension d'une semaine pour le compléter, en raison de plusieurs imprévus qui nous ont mis des bâtons dans les roues.