Log680 - Laboratoire 3

02 aout 2024

Robyn Girardeau

Nicolas Patenaude

Introduction

Ce troisième laboratoire constiste en plusieurs parties:

1. Planification du travail

- Ajouter les tâches aux Kanban
- Attribuer/Assigner les tâches à Nicolas et Robyn

2. Déploiements Kubernetes

- Déploiements Kubernetes
- Déploiement de Metrics API sur Kubernetes
- Déploiement de Oxygen OS sur Kubernetes
- Déploiement du CronJob sur Kubernetes

3. Visualisation sur Grafana

- Connexion entre Grafana et la base de données
- Création d'un dashboard Grafana avec les métriques HVAC
- Création d'un dashboard Grafana avec les métriques de processus

Répartition des tâches

La répartition des tâche était très simple. Robyn à accompli les tâches en lien avec Kubernetes suivantes:

- Déploiements Kubernetes
- Déploiement de Metrics API sur Kubernetes
- Déploiement de Oxygen OS sur Kubernetes
- Déploiement du CronJob sur Kubernetes

Tandis que Nicolas à accompli les tâches suivantes en lien avec Grafana:

- Connexion entre Grafana et la base de données
- Création d'un dashboard Grafana avec les métriques HVAC
- Création d'un dashboard Grafana avec les métriques de processus

La répartion des tâches fut équitable. Comme mentionné dans nos weekly stand-ups, nous n'avons pas renconté d'obstacles ou de bloquants

Arboresence du projet (inchangée depuis lab2)

├── .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 dépot 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.

Implémentation de notre déploiement continu

Nous utilisons le paramêtre imagePullPolicy: Always afin de faire en sorte que chacunes des modifications sur notre branche principale enclenche le téléchargement de l'image la plus à jour.

L'api de metrics est disponible à partir de l'addresse suivante: http://146.190.191.28/user01eq15/metrics/

La commande `kubectl rollout restart deploy YOUR-DEPLOYMENT` nous permet de redémarrer l'application avec la dernière version manuellement.

Consultation des métriques

Deux dashboards ont été créé dans graphana.

Le premier dashboard contient les métriques, il est accessible au lien suivant: http://159.203.54.78/d/cdt90zchr1erkd/metrics?orgId=111

Le deuxième dashboard contient les données du HVAC, il est accessible au lien suivant: http://159.203.54.78/d/ddt8ywjiyvabkf/oxygencs-dashboard?orgId=111&refresh=5s

Conclusion

Ce dernier laboratoires s'est très bien déroulé. Malgré le fait que nous avons eu une semaine en moins pour le compléter en raison de l'extension qui nous à été accordé pour le deuxième laboratoire.