|  |
| --- |
| Rapport : Virtualisation  2023 |
|  |
| 26 mars  Rédigé par : Rémy CHANTHAVONG Dara DULON Alexandre FRANCINEAU |



# Table des matières :

# Introduction

# Prérequis

# Fonctionnalités

# Conclusion

# Qwiklabs

# Introduction

L’objectif du projet à été de créer une page web PHP capable d’afficher des données fournit par une base de données MySQL, ainsi que de déployer l’ensemble du projet sous Kubernetes.

# Prérequis

Pour réaliser ce projet nous avons eu besoin d’avoir Docker d’installé, ainsi que Kubernetes/minikube d’installé sur Windows.

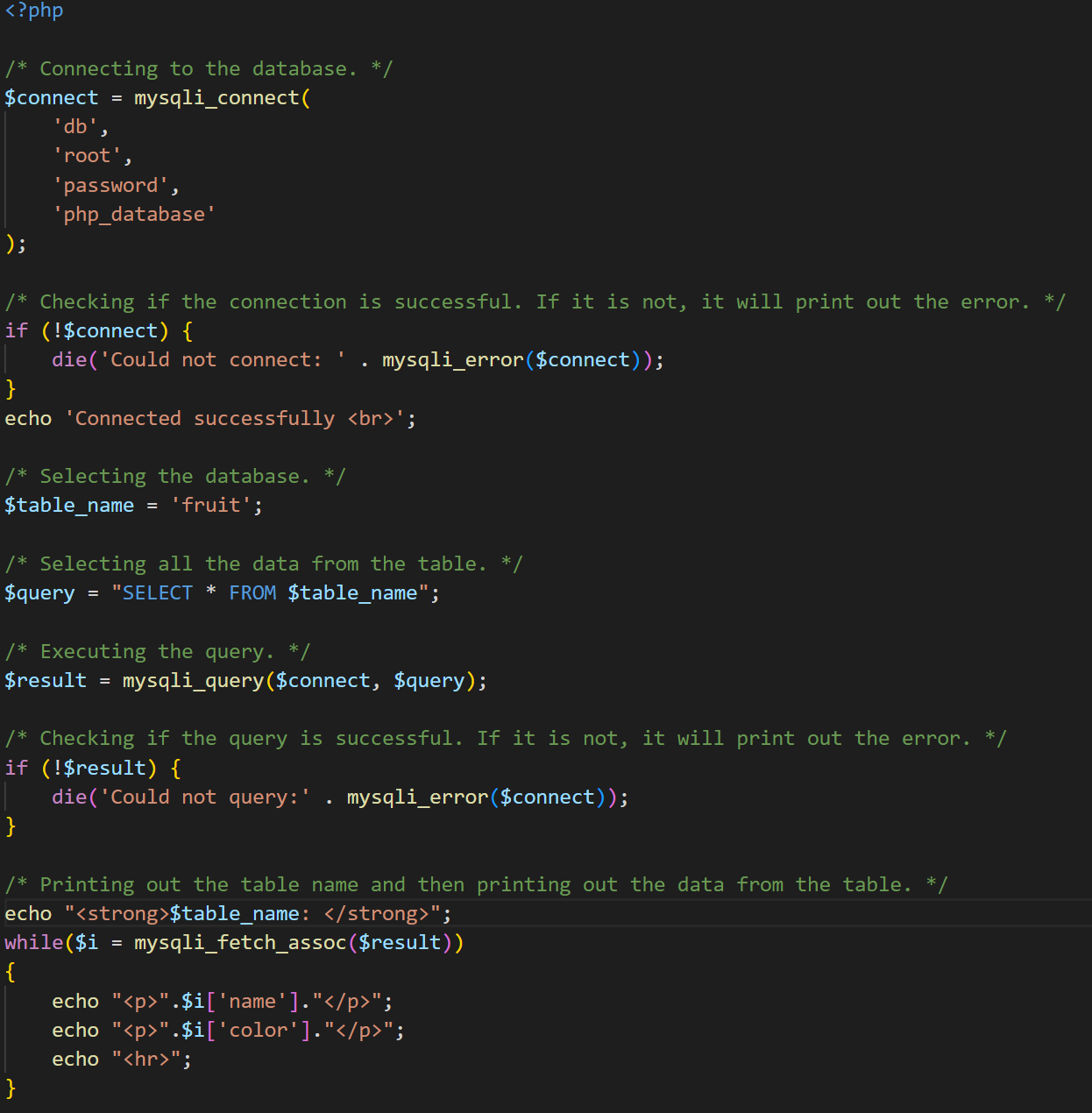
# Fonctionnalités

Nous avons choisi d’utiliser le langage PHP afin de réaliser notre premier service qui consistera à afficher les données d’une table dans notre base de données MySQL.

Tout d’abord nous avons créé 2 Dockerfile, Dockerfile.db (qui contiendra les informations concernant notre base de données MySQL) et Dockerfile.php\_web (qui contiendra les paramètres de notre conteneur PHP).

Ces deux Dockerfile sont utilisés par un fichier docker-compose.yml afin de paramétrer simplement plusieurs conteneurs.

On vient par la suite créer notre index.php, qui servira de visuel pour notre base de données :

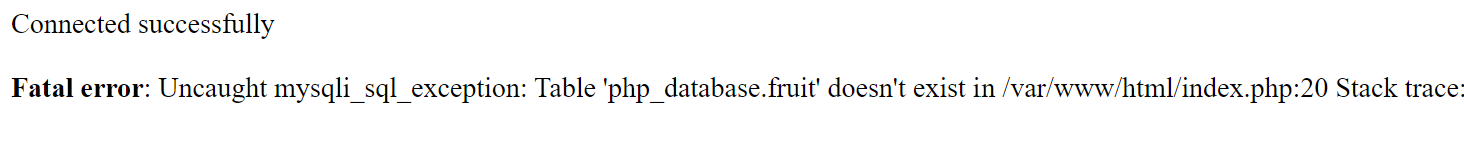


On exécute ensuite la commande : « docker-compose up –build » afin de créer les images et créer les conteneurs correspondants.

Par la suite on peut envoyer les images sur DockerHub pour les utiliser sous Kubernetes.

Par la suite on a pu démarrer le cluster Kubernetes, et ainsi déployer les images Docker dans le cluster en utilisant des fichiers .yml. Dans ces fichiers on spécifie notamment l’image Docker sur DockerHub que l’on doit utiliser, ainsi que les services à créer. Et ensuite on vient utiliser la commande « kubectl apply -f \*.yml ».

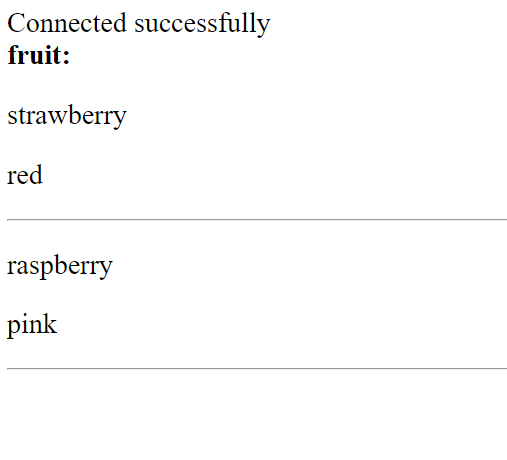
On a ensuite testé l’accès à la page PHP depuis minikube :



On s’aperçoit que l’on a bien accès à notre page web, cependant on se retrouve avec une erreur, ce qui est normal puisque la table php.database.fruit n’existe pas encore.

On vient donc dans le terminal du pod de la base de données afin d’y créer notre table et y inscrire quelques données.

Au final on obtient ceci :



# Conclusion

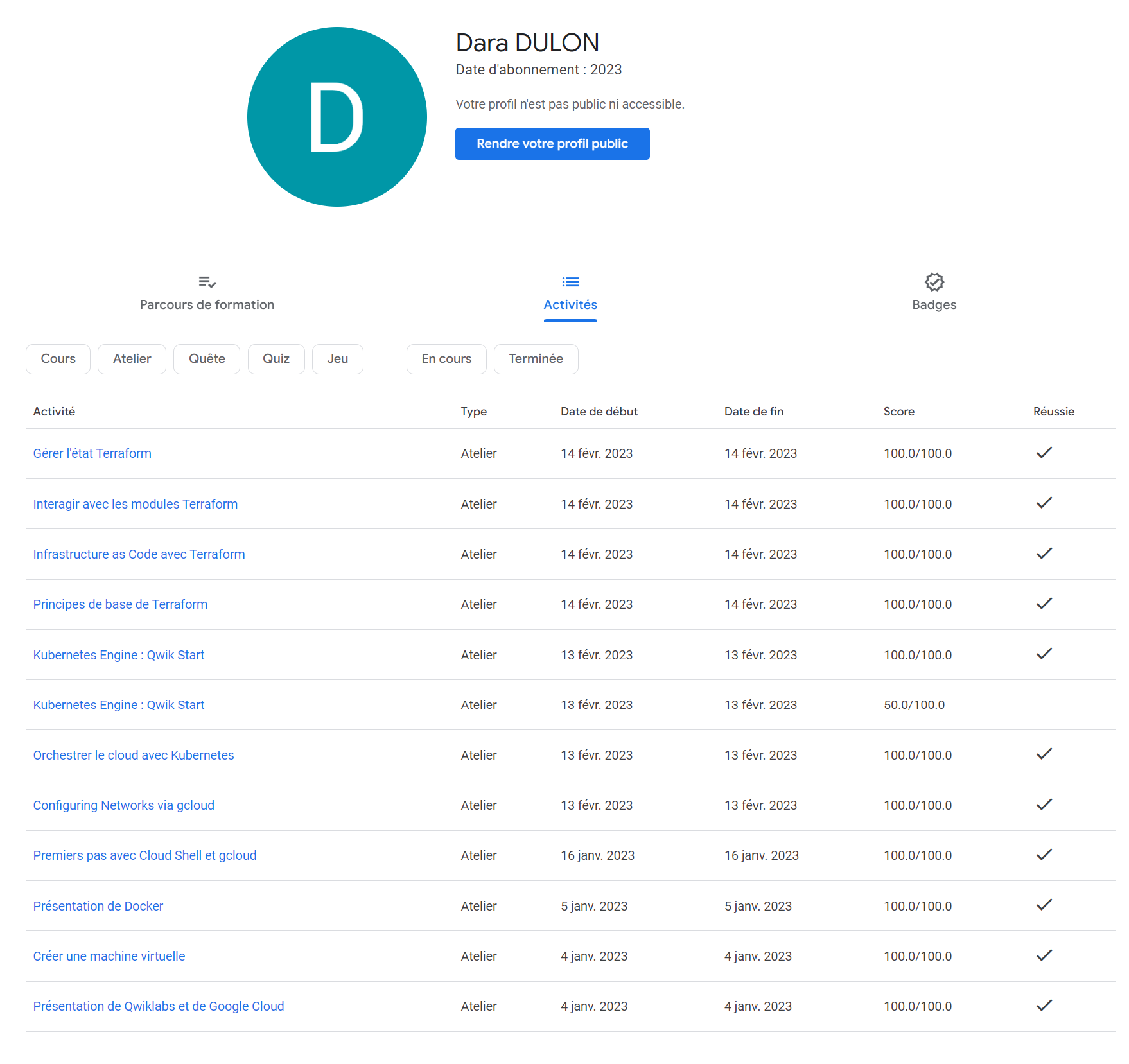
# Entre notre projet, qui a amélioré notre compréhension des microservices et du langage de programmation PHP, et les Qwiklabs, qui nous ont beaucoup appris sur Docker et Kubernetes. Nous avons appris la valeur de certaines tâches dont nous n'avions aucune idée de l'importance avant de suivre ce cours, ce qui a été très instructif.

# Qwiklabs

# Qwiklabs de Rémy CHANTHAVONG :

# 

# Qwiklabs de Dara DULON :



# Qwiklabs d’Alexandre FRANCINEAU :

