

CONTENEURISATION DOCKER #1

HASAJ Andi, LACI Senard, GROLLEAU Florian - 2A2

Groupe N°8

Déroulé de ce document

I – Setup de l’application

Guide d’installation et de démarrage de l’application.

II - Se servir de l’application et son fonctionnement

Guide d’utilisation et de vérification du fonctionnement de l’application.

III – Architecture

Survol et présentation de l’architecture de l’application.

I – Guide de démarrage et installation

Installation des modules : Pour installer les modules et utiliser l’application, rendez-vous sur le [repository du projet](#) pour visualiser les packages. Vous recherchez deux images en particulier, ‘*frontend-le-catering*’, ‘*backend-le-catering*’.

Pour se faire, vous avez deux commandes à entrer :

↪ *docker pull ahasaj/backend-le-catering:latest*
↪ *docker pull ahasaj/frontend-le-catering:latest*

Pour démarrer le projet, pensez bien à suivre l’ordre de démarrage suivant :

(BDD) → Backend → Frontend

La raison de ceci est car lors du démarrage, chaque conteneur va chercher ses dépendances et vérifier qu’elles sont présentes et opérationnelles. La partie Frontend a besoin de la partie Backend (qui a besoin de la base de données, celle-ci est normalement initialisée de base avec la partie backend en JAVA).

Vous exécuterez donc ces commandes dans l’ordre

↪ *docker run backend-le-catering*
↪ *docker run frontend-le-catering*

Une fois ceci fait, entrez la commande *docker ps -a* pour vérifier que toutes les images se sont lancées et que le site est prêt à l’emploi.

Si tout est fait, vous devriez normalement être prêt à utiliser le projet en vous rendant sur <http://localhost:5000>.

*NOTE : Si vous avez accès au fichier ‘docker-compose.yml’, vous pouvez également installer le projet directement en entrant uniquement la commande suivante : **docker compose -f docker-compose.yml up --build -d***

II – Guide d'utilisation

Cette partie est dédiée à l'usage du site internet et fera une présentation brève. Vous devriez normalement, si l'installation s'est passée correctement sans accrochage ou panne ou erreur imprévue, vous arriverez sur cette page.



P.S. Veuillez ne pas faire attention à la résolution de l'image d'arrière plan. Celle-ci s'est montrée bien moins haute qu'originalement indiquée.

On reprend ici le thème d'un site web dédié à un restaurant factice nommé "The Catering", de plus, si une similarité avec une entité actuellement existante (nom similaire à un autre restaurant, article(s) du menu existant sur d'autres cartes), il ne s'agit ici que d'une coïncidence. De plus, la page de base a été récupérée sur le site internet w3schools, cela est fait mention dans le footer de la page.

Cette page a tout ce qui semble être de plus banal, une présentation du restaurant et son histoire, un menu copieux avec des images, mais une partie du site nous intéressera ici. Il s'agit du formulaire de réservation.

Contact

We offer full-service catering for any event, large or small. We understand your needs and we will cater the food to satisfy the biggerst criteria of them all, both look and taste. Do not hesitate to contact us.

Catering Service, 42nd Living St, 43043 New York, NY

You can also contact us by phone +1 (806) 555-1337 or email catering@catering.com, or you can send us a message here:

Name

How many people

jj/mm/aaaa --:--



Message \ Special requirements

SEND MESSAGE

Ce formulaire est connecté à une base de données SQL. Entrez une réservation de votre choix et celle-ci devrait être enregistrée.

Voici un exemple de à quoi cette réservation peut ressembler dans la base de données.

name	nb_customers	date	message
John Doe	4	2026-02-25 15:48:39	Can you please place us near the windows out front ? Thanks in advance.

Invite de commande utilisant MariaDB.

Vous pouvez vérifier en accédant à la base de données et en entrant ces deux commandes.

1. **USE restaurant;**
2. **SELECT * FROM contact;**

III – Revue architecturale et logicielle

Cette partie est dédiée à la présentation de l'architecture du projet.

1. Répartition des fichiers

Les fichiers ont été répartis ainsi :

```
\|
|| Documentation -> Ce fichier que vous êtes actuellement en train de lire
||
|| le-catering
|| \
|| | backend
|| | \
|| | | java
|| | |
|| | | ... -> App.java
|| | | ressources -> application.properties
|| | |
|| | | sql
|| | |
|| | | restaurant.db
|| | |
|| | | target
|| | |
|| | | \
|| | | | classes relatives au code du backend
|| | |
|| | | \
|| | | |
|| | | | frontend
|| | |
|| | | \
|| | | | code et ressources du site web
|| | |
|| | | \
|| | | |
|| | docker-compose.yml
|| |
|| README.md
```

Pourquoi cette sélection ? Car, pour nous, elle semble intuitive et simple à comprendre, les parties sont bien sectionnées entre elles et

sont rapidement compréhensibles pour quelqu'un qui met les mains dans ce dossier, de plus, un arrangement tel nous permet d'économiser de la place sur la machine hôte en combinant la partie Backend et BDD.

Néanmoins, ce choix se fait donc à une concession, qui est que en combinant la partie Backend et BDD, si une erreur survient sur un des deux composants, la partie Backend et BDD seront tous deux affectés, impactant donc notre stabilité en cas de problème.

2. Choix des services

On a choisi pour le service d'hôte de serveur Web : NGINX. Ce choix est très simple, il s'agit d'un logiciel éprouvé et simple à prendre en main tout en étant l'un des plus populaires donc celui-ci s'accompagne d'une documentation complète et reste tout de même performant.

Pour le logiciel de gestion de la base de données, on a choisi PostgreSQL, de même ici, on a fait ce choix puisqu'il s'agit d'un service reconnu, bien documenté et facile pour nous de prendre en main, un autre choix notable qu'on aurait aussi pu choisir aurait été MySQL.

3. Configuration des volumes

On a créé un seul volume ici, il est dédié à la base de données pour la persistance des données, aucun autre volume existe pour la simple raison qu'il serait inutile de dédier d'autres volumes pour la partie frontend et la partie code de la backend. Donc vraisemblablement, seule la BDD en aura besoin.

4. Configuration des réseaux

Il y a deux réseaux distincts présents dans ce projet, un qui relie la partie frontend au backend, et le deuxième qui relie le backend à la base de données. Ce choix a été fait pour des raisons de sécurité, en effet, il est préférable de ne pas connecter le frontend directement à la BDD. Le but étant d'isoler les composants au strict nécessaire.

[Remonter en haut de page](#)