



# Spécifications projet : Microservices-M2



**Nom:**

**Prénom :**

Ci-joint la liste de spécifications de votre projet de deuxième année du module Microservices.

## **Composants techniques de bases (4points) :**

- Votre projet contient un frontend avec a minima deux routes
- Votre projet contient un backend avec une base de données dans un container spécifique. Vous expliquerez le choix de votre technologie de base de données (relationnelles, ou non relationnelles)
- Votre base de données est fonctionnelle et composée d'au moins une table/ collection ainsi qu'un schéma explicite.
- Architecture docker :
  - Votre projet contient au moins 2 Dockerfile et un fichier docker-compose.yml
  - Votre projet contient un docker network
  - Vous avez push au moins 2 image docker sur un container registry (type docker-hub)
- Votre projet est accessible en ligne publiquement sur un gestionnaire de version type GitHub, bitbucket, gitlab...

## **Architecture & Clean Code (2points)**

- Vous avez schématisé l'architecture de votre projet sous format vectoriel ([draw.io](#) ou autre web tools) avec notamment :
  - les différents composants/services de votre application: liens entre ces différents composants/services : protocoles et requêtes types (POST,GET,... in curl/json format)
  - les ports exposés sont mis en valeur (côté client et coté backend)
- Votre README.md à la racine de votre GitHub contient les explications nécessaires pour build et run votre projet ainsi que des screens shot de vos fonctionnalités ainsi que des critères demandés.

## **Déploiement & production (4points)**

- Votre projet dispose d'un service de reverse proxy type nginx ou traefik.
- Vous avez mis en place SSL et HTTPS avec des certificats Let's Encrypt.
- Votre projet dispose d'une documentation (auto générées vous le voulez type OpenAPI/formerly Swagger Specification) documentant les différents endpoints/ fonctionnalités importantes du projet.
- Vous avez assurer le monitoring en temps réel de vos containers et pouvez partager un dashboard de monitoring via une URL externe (graphana, prometheus, weavescope...)
- Votre solution dispose d'un système de gestion de files d'attente ou de push notification

### Tests (4points)

- Vous avez a minima des tests container (bash ou healthcheck) de tests qui :
  - test la santé de vos conteneurs
  - test le ping de vos application (front / back)
  - test du bon fonctionnement de votre base de données
- Vous avez testé unitairement a minima 2 composant de votre code base
- Vous avez mis en place des tests de montée en charge simple (type <https://locust.io/>) de votre services
- Vous avez un parcours de test end to end (<https://playwright.dev/python/docs/intro> en python <https://www.cypress.io/> en js)

### Continus Integration/Deployment with github (4points)

- Vous a minima 2 branches (main et dev) pour organiser votre travail et tester vos workflow sur les évènement de pull request, push and merge
- Vous avez testé vos GitHub workflow en local avec l'outil *act* (screen shot à l'appui)
- Vous avez mis en place les github workflows suivant dans votre projet :
  - Run and test your frontend server
  - Run and test your backend server
  - Build docker image for your backend server
  - Build docker image for your frontend server
  - Publish the docker image of your frontend and backend on your **public** dockerhub repository
  - Add build button (to be more pro 😎)
- Vous avez automatiser votre déploiement sur un cloud server type Cloudron/ Coolify, mettez un screen shot dans vote documentation.

### Contrôle continue : suivi des push git : [2push/demi journée] (2points)