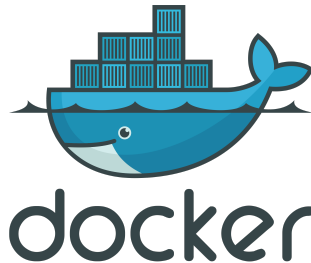


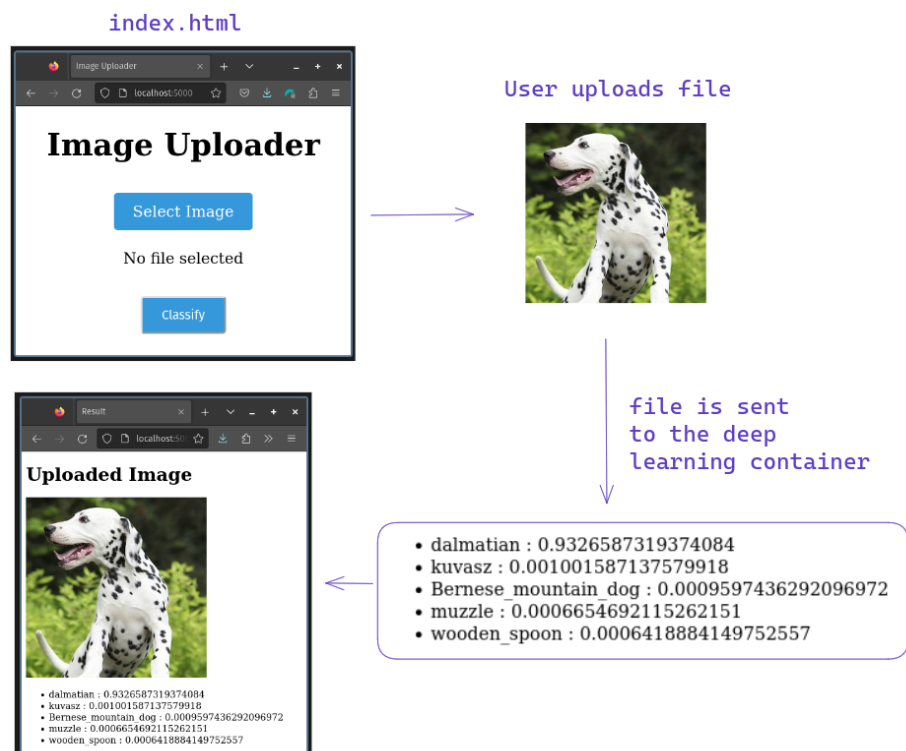
Introduction à



VIII. Mini projet

L'objectif de ce mini projet est de créer une mini application web conteneurisée permettant de classifier des images en utilisant un modèle de deep learning existant.

- Une application web interagit avec l'utilisateur pour lui permettre d'uploader une image.
- Cette image sera envoyée à un second container capable de classifier le contenu de l'image
- Le résultat de classification est envoyé au premier container
- Le premier container reçoit le résultat et l'affiche.



Itération 1 : Créer une application web écrite en **Flask** et la conteneuriser.

Itération 2 : Créer une stack avec **docker-compose** qui permettra de créer un second container à partir d'une image publique disponible sur Docker Hub.

Itération 1 : application Flask

1. La première brique élémentaire de ce projet est de créer une mini application web renvoyant une page HTML index.html lors d'une requête GET sur **localhost:5000/**

Ressource :

- Flask : [Quickstart — Flask Documentation \(1.1.x\)](#)
- HTML : [HTML Tutorial](#)

2. Conteneurisez cette application.

- Écrivez un dockerfile permettant de construire une image de votre projet
- Exécutez votre projet pour vérifier que tout fonctionne.

Remarques :

- Pensez à mapper volumes et ports
- Pensez à choisir une image adéquate sur Docker Hub (ne réinventez pas la roue)

Ressource : [How to Dockerize a Flask Application](#)

3. Modifiez la page index.html afin de pouvoir uploader une image.

Ressource : [How To Create a File Upload Button](#)

4. Créer une route POST dans l'application Flask qui intercepte l'image qui est uploadé grâce au formulaire HTML et enverra le fichier au (futur) second container (vous pouvez la renvoyer sur la même url pour le moment)

Ressource : https://jdhao.github.io/2020/04/12/build_webapi_with_flask_s2/

Interlude : Etude de docker-Compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

Site de Docker

<https://docs.docker.com/compose/>

Docker-Compose est un outil permettant de définir des **applications multi-containers**, appelées **Stacks** (pile en français). Vous aurez besoin d'écrire un fichier **yml** décrivant la manière dont sont créés et agencés les containers les uns par rapport aux autres, et comment ils communiquent (qui doit être créé en premier? qui dépend de qui? etc.)

Un gros avantage de docker compose est de pouvoir aller chercher très facilement des images dont nous avons besoin et de les faire fonctionner ensemble : j'ai besoin d'une base de données postgres, d'un serveur NGinx et d'un modèle de Deep Learning exposant une API → pas de soucis, je n'ai qu'à aller récupérer ce qui a déjà été fait par la communauté, sans avoir besoin de tout installer chez moi, et sans briser ma config machine ! C'est un peu comme faire ses courses en ligne !

Etude de Docker Compose :

Commencez par aller étudier les ressources sur **Docker-Compose** et essayez de reproduire les exemples donnés. Vous devriez maintenant être autonomes pour appréhender la doc, et les tutoriels.

Ressources :

- Doc officielle : [Docker Compose overview](#)
- Baeldung (jusqu'au 4 inclus) : [Introduction to Docker Compose | Baeldung](#)
- Trouvez les ressources qui vous conviennent, elles sont nombreuses.

Itération 2 : Utilisation des modèles de deep learning publiés par IBM.

Maintenant que vous êtes des expert.e.s en docker-compose passons aux choses sérieuses !

1. Vous trouverez ici une image déjà prête : [codait/max-inception-resnet-v2 - Docker Image](#)
2. Essayez de l'utiliser "à la main" pour le moment. **Regardez la doc**, vous pouvez lancer le container seul pour l'essayer et comprendre quelles sont **API endpoints**, et les formats d'entrées et de sorties.
3. Essayez de faire communiquer les deux containers pour arriver au comportement attendu.

Ressources :

1. Communication entre containers avec Docker Compose : [Networking in Compose | Docker Docs](#)
2. Infos sur les API Rest :
 - [What is REST – A Simple Explanation for Beginners, Part 1: Introduction | by Shif Ben Avraham | Extend | Medium](#)
 - [A Beginner's Tutorial for Understanding RESTful API \[MLSDev\]](#)

Extra Bonus :

De nombreux autres images sont disponibles sur le compte de **codait** : [codait's Profile | Docker Hub](#)

Introduction à Docker

KIT APPRENANT

Si vous êtes encore motivé.e.s vous pouvez en essayer d'autres : Artistic Style Transfer, Optical Character Recognition, Text Summarizer, ..