

## AA: Cloud & Edge Computing

### Enoncé du projet

#### I. Introduction :

Comme annoncé en séance de cours, ce projet s'appuiera sur les connaissances acquises durant les séances de cours et TP du module « Cloud & Edge Computing »

Les projets seront réalisés par groupe de deux avec les échéances suivantes :

- Date de remise du projet (rapport d'environ 20 pages + code + manuel) : le **08/06/2024** via Moodle
- Mode de présentation du projet : en présentiel (Ho. 25)
- Date de présentation du projet : le **14/06/2024** ou chaque groupe aura **15 minutes** au maximum pour présenter son projet suivi de **5 à 10 minutes** de questions.

**Note 1 :** lors de la présentation de vos projets, vous pourrez avoir des questions liées à la théorie vue au cours.

L'ordre de passage :

- **08h15 – 08h40** : Groupe 01 (Eléonore Vandendaele et Boris Wilmart)
- **08h40 – 09h05** : Groupe 02 (Nicolas Badot et Loïc Barbier)
- **09h05 – 09h30** : Groupe 03 (Noa Foucoux et Nicolas Sournac)
- **09h30 – 09h55** : Groupe 14 (Gerard pongo Elonge)
- **09h55 – 10h20** : Groupe 05 (Marine Bodson et Hugo Venturoso)
  
- **10h50 – 11h15** : Groupe 06 (Arnaud Moreau et Cyril Moreau)
- **11h15 – 11h40** : Groupe 07 (Samuel Sauvage et François Vion)
- **11h40 – 12h05** : Groupe 08 (Thomas Bernard et Virgil Surin)
- **12h05 – 12h30** : Groupe 09 (Alessandro Spinosi et Nicolas Valloi)
  
- **13h30 – 14h00** : Groupe 10 (Cyril Kaisin, Jules Pimont et Van-Minh Christophe LE)
- **14h00 – 14h25** : Groupe 11 (Abdelouahed Alla et Mael Njopmou Nankap)
- **14h25 – 14h50** : Groupe 12 (Kévin Dubrulle et Lara Kacikowski)
- **14h50 – 15h15** : Groupe 13 (Lionel Daghan et Arsene Mujiyabwami)
- **15h15 – 15h40** : Groupe 04 (Rodrigue Deghorian et Hugo Lorenzini)

## II. Enoncé du projet :

Le but du projet est d'héberger une application d'indexation et recherche multimédia sur ressources Cloud (ou Edge). Nous vous proposons d'utiliser vos machines virtuelles créées dans le cadre du cours de « Cloud and Edge Computing » pour héberger vos applications. Le projet comprend deux parties :

### • II.1. Partie 01 : application d'indexation et recherche multimédia :

L'objectif de cette partie est de développer un moteur de recherche exploitant # descripteurs, il faudra :

1. Indexer la base de données avec les descripteurs de votre choix. Si plusieurs descripteurs sont choisis, il faudra donner la possibilité de les combiner ;
2. Réaliser la recherche en donnant la possibilité de choisir la fonction de calcul de similarité (Euclidéenne, Corrélation, Chi-square, Bhattacharyya, Brute Force Matcher, Flann, etc.) ;
3. Afficher le Top20 et Top50 pour les images requêtes ;
4. Calculer le Rappel (R), Précision (P), Average Precision (AP), Mean Average Precision (MaP) et R-Precision

Le code Python et solution du moteur de recherche sera fourni. Votre travail sera focalisé sur l'hébergement de ce moteur de recherche. Nous vous invitons à tester et bien analyser le code fourni afin d'identifier le meilleur moyen de déployer la solution sur le Cloud

### II.1. Partie 02 : hébergement de l'application sur ressource Cloud ou Edge :

L'objectif de cette partie est d'héberger votre application de recherche multimédia (de la partie 1) sur une ressource Cloud ou Edge afin d'offrir un service sous forme de Software As A Service « **SAAS** ». Nous vous proposons de suivre ces six (06) étapes :

1. **Indexation « extraction de caractéristiques » en local** : en raison des performances limitées de votre machine virtuelle (pas de GPU), nous vous proposons de sélectionner votre meilleur modèle (s) et fichier de caractéristiques d'images avant de les copier vers votre machine virtuelle. La phase d'indexation ne doit donc pas être hébergée sur ressource cloud.
2. **Test et configuration de votre application de recherche sur ressource Cloud** : ici, il faudra installer et configurer votre machine virtuelle afin de tester votre application (partie 1) sur la ressource Cloud.

3. **Génération de l'image Docker regroupant les fonctionnalités de votre application** : ici, il faudra créer un Dockerfile regroupant les instructions nécessaires pour faire fonctionner votre application. Notons que votre image devra gérer :
  - a. **En entrée** : une image requête ;
  - b. **En sortie** : les indices des images les plus similaires + la courbe de Rappel/Précision.
4. **Développement d'une page Web pour faciliter l'accès au service SAAS** : ici, il faudra développer une page Web (avec [flask](#) ou [django](#) voire [php](#)) permettant de :
  - a. Afficher les informations des développeurs du projet & description/fonctionnalités de votre application ;
  - b. Lancer l'application de la recherche à l'aide de boutons, labels, etc.
  - c. Afficher les résultats de la recherche : images similaires (avec taux de similarité) + courbes R/P
  - d. Compléter les tableau 1 et tableau 2 (à inclure dans votre rapport)
5. **Configuration d'accès** : configurer l'accès à votre service à l'aide de votre @et numéro de port au choix
6. **Personnaliser votre site** : selon votre imagination en incluant une page de connexion
7. **Flexibilité d'utilisation** : permettre à l'utilisateur de choisir la méthode d'indexation et recherche

#### Facultatif :

8. **Combinaison de services** : utiliser docker-compose pour combiner les services
9. **Cybersécurité** : analyser votre site en termes de sécurité avant de l'améliorer dans ce sens
10. **Mise à l'échelle** : programmer la mise à l'échelle en fonction de la charge de votre moteur (**TP4 facultatif sur Moodle**).

- **Note 3** : pour la **partie 2**, on pourra augmenter les capacités de mémoire (stockage et RAM) et de calcul selon vos besoins. Ceci vous permettra d'installer tous les outils nécessaires.

La figure 2 illustre un exemple d'hébergement de l'application de recherche d'images en utilisant une image Docker et une page Web développée à l'aide de php et html. Vous pouvez également visualiser cette [vidéo](#) pour avoir une idée simple et claire du travail attendu.

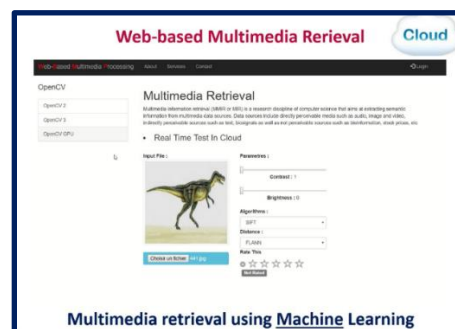


Figure 1: exemple d'hébergement d'application de recherche multimédia

### III. Quelques liens intéressants :

- Exemple d'hébergement d'une application **C++** de traitement d'images avec Docker et php : voir ce [lien](#).
- Exemple d'hébergement d'une application **python** de classification d'images « Deep Learning » avec Docker et php : voir ce [lien](#).

### IV. Séances Projet : réservées à l'horaire du cours « ML & DL for Multimedia Retrieval » :

a. 25/04 de 10h30 à 12h30

b. 02/05 de 08h15 à 10h15

### V. Contact: Sidi Ahmed Mahmoudi, Nourredine Bendjelloul et Aurélie Cools