

# Applications of Big Data - Part. 2

Vincent DOMINGUES

[vincent.domingues@intervenants.efrei.net](mailto:vincent.domingues@intervenants.efrei.net)

<https://www.linkedin.com/in/vincentdmg/>



# Objectifs

Déployer, au travers d'une API, un modèle entraîné de prediction en utilisant la philosophie DevOps sur un cloud provider.



# Notation du module

Par groupe, de 4 ou 5 personnes, vous devrez présenter votre projet pendant une soutenance de 20 min et fournir tous les livrables demandés par mail.

## Déroulement de la soutenance par groupe

15 min de présentation

5 min de questions

## Livrables

- URL repository Github avec tous les éléments (code, Dockerfile, configuration workflow Github)
- Nom de l'image docker sur Azure Container Registry (ACR)
- Endpoint API Azure Container App
- Rapport qui présente, étape par étape, les choix techniques, les commandes utilisées, la stratégie de test de charge accompagnée de vos observations et les difficultés rencontrées si vous n'avez pas pu terminer.

# Template

Dataset: <https://archive.ics.uci.edu/ml/datasets/iris>

Algorithme: scikit-learn KNN (K-nearest neighbor)

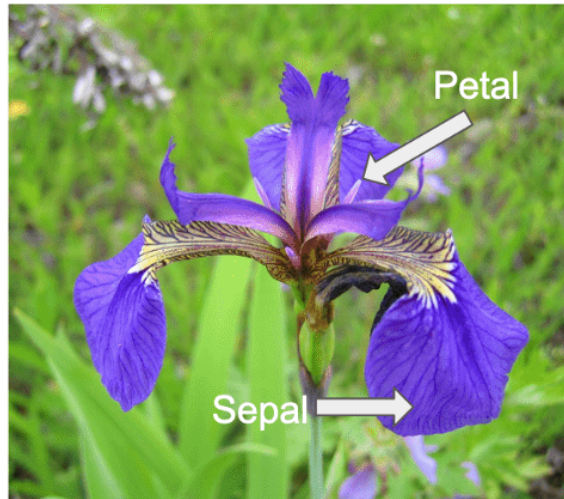
Language de programmation: Python

Cloud provider: Azure (Resource group: **ADDE92-CTP**)

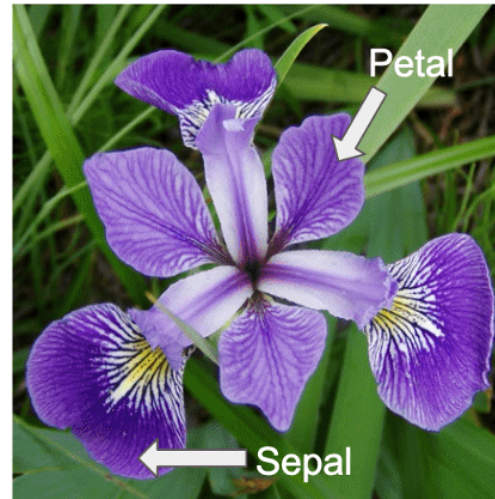
Organisation Github: efrei-ADDE92-CTP

Azure Container Registry: efreiprediction.azurecr.io

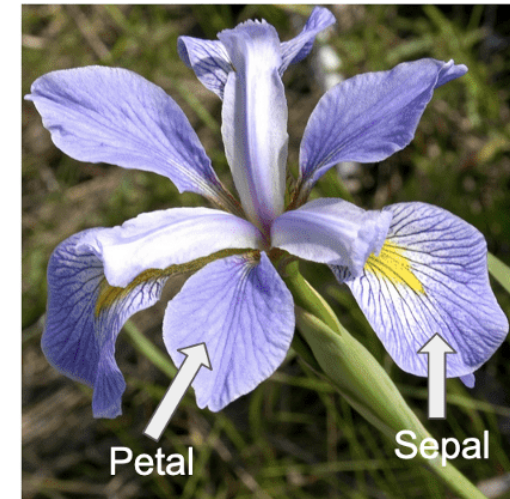
*Iris setosa*



*Iris versicolor*



*Iris virginica*



# Etapes

- Training du modèle (Iris avec scikit-learn KNN) et export (joblib.dump)
- Intégrer son modèle entraîné dans une API
- Packager son API dans une image Docker en local
- Publier son code sur Github
- Configurer la pipeline de déploiement avec les éléments suivants
  - Construire l'image Docker
  - Publier l'image Docker sur Azure Container Registry (ACR)
  - Déployer sur Azure Container App
    - Configurer l'autoscaling en utilisant comme paramètre le nombre de requêtes en simultané
- Test de charge avec l'outil de votre choix et observer l'autoscaling
- Ajouter un endpoint `/metrics` en utilisant la librairie `prometheus-client` puis mettre a disposition une/des metriques d'utilisation de l'API (exemple: counter avec le nombre de calls)

# Dépôt livrables

A envoyer par mail au plus tard 24h après la fin de la soutenance à

[vincent.domingues@intervenants.efrei.net](mailto:vincent.domingues@intervenants.efrei.net) avec comme objet:

**[PROJET-BIGDATA-GROUP\_NUMBER] NOM1 NOM2 NOM3 NOM4**

*Pensez à bien renseigner le nom de chaque membre du groupe.*

## Bonus

- Utiliser un linter pour Dockerfile dans la pipeline de déploiement pour s'assurer de sa cohérence
- Mettre en place une stack prometheus local pour scraper et stocker vos metriques exposées

# Credentials a disposition sur l'organisation Github

**AZURE\_CREDENTIALS** service account pour s'authentifier avec l'API Azure

**REGISTRY\_LOGIN\_SERVER** lien de la registry ([efreiprediction.azurecr.io](https://efreiprediction.azurecr.io))

**REGISTRY\_USERNAME** username de la registry

**REGISTRY\_PASSWORD** password pour la registry

**RESOURCE\_GROUP** resource group Azure (ADDE92-CTP)