

# Pipeline de traitement de données pour le Cloud

UPEC / EPISEN 2025/26 S1

Evaluation finale sous la forme d'un projet

- Projet à présenter au cours #6, le 18/12/2025
- A réaliser seul ou en binôme
- Sujet du projet
- Exigences
- Grille d'évaluation
- Choix des équipes

# Sujet du projet

---

- Vous travaillez pour un intégrateur spécialisé dans les services analytiques basés sur le Cloud AWS et votre devise est « *Do, not tell* »
- Vous devez réaliser une preuve de concept (PoC) pour un nouveau client, afin de le convaincre d'utiliser vos services et de gagner un premier projet
- Vous allez réaliser un pipeline de traitement de données dans AWS pour illustrer un cas d'usage de votre choix et impressionner votre client (effet « Waouh »)
- Vous pourrez utiliser les services AWS vus en cours et dans les labs, ou d'autres services AWS de votre choix (disponibles dans le « Student Lab » depuis le portail AWS Academy)
- Pendant la restitution de votre PoC devant votre client (devant la classe), vous devrez présenter le cas d'usage, l'architecture et une démonstration convaincante du pipeline, ainsi qu'une analyse du coût total de la solution dans AWS (coût des services Cloud AWS proposés).

# Exigences du projet

---

1. Vous pourrez travailler seul ou en binôme – la liste des “équipes” sera disponible sur Teams (fichier Excel)
2. Si vous décidez de travailler en binôme, le travail doit être réparti équitablement, et chaque membre de l’équipe devra présenter pendant la moitié du temps de restitution
3. Votre cas d’usage doit être original – deux “équipes” avec le même cas d’usage verront leurs notes divisées par 2
4. Vous présenterez votre cas d’usage d’un point de vue fonctionnel (contexte métier de votre client) et selon les 5 caractéristiques en « v » : volume, vitesse, variété, véracité et valeur

# Exigences du projet

5. Vous exposerez l'architecture détaillée du pipeline, en vous appuyant sur un diagramme d'architecture – vous pourrez vous inspirer des diagrammes vus en cours et dans les labs, ou dans la documentation AWS
  - Exemple d'outil pour réaliser des diagrammes d'architecture avec modèles pour AWS : <https://app.diagrams.net> (draw.io)
  - Icônes des services AWS : <https://aws.amazon.com/fr/architecture/icons/>
6. Votre pipeline devra comporter les 4 phases d'un pipeline de données, que vous présenterez selon le modèle vu en cours :



# Exigences du projet

---

7. Votre pipeline devra être totalement ou en grande partie automatisé, au niveau de son déploiement dans AWS et de son exécution
8. Pour automatiser le déploiement du pipeline, vous utiliserez Terraform ou le service CloudFormation d'AWS (vous pourrez vous baser sur les exemples vus en cours et dans les labs)
9. Votre restitution devra comporter les éléments suivants :
  - Une présentation détaillée du cas d'usage et du pipeline de traitement de données dans AWS, avec son architecture
  - Une démonstration du pipeline, qu'elle se fasse en direct ou bien qu'elle soit enregistrée (vidéo)
  - Un référentiel Git avec toutes les ressources du pipeline (présentation, diagrammes, code source, fichiers d'installation...).

# Grille d'évaluation du projet

---

- Le projet sera évalué sur la base de votre présentation au cours #6
- Durée de votre présentation : environ 13 minutes, suivies de 5 minutes de questions
- Critères d'évaluation du projet :
  1. Une présentation maîtrisée et convaincante – 20%
  2. Un cas d'usage clair et original – 20%
  3. Une architecture bien expliquée et conforme aux exigences – 20%
  4. Une démo convaincante, incluant tous les aspects du pipeline de données et son automatisation – 20%
  5. Le respect de toutes les exigences du projet – 20%

# Grille d'évaluation du cours

---

- Contrôle continu : 50%
- Projet : 50%

# Choix des équipes pour le projet

---

- Fichier Excel dans Teams avec la liste des équipes
- Vous aurez jusqu'au cours #5 (4/12/2025) pour déclarer votre équipe.



# Cas d'usage des data pipelines

Ce ne sont que des exemples,  
à vous de faire preuve  
d'originalité !

- Analyse de logs
- Analyse de données issues de capteurs / objets connectés
- Analyse de données des réseaux sociaux (ex. tweets, données de Spotify...)
- Analyse de données financières (ex. stock ticker)

...

# Ressources pour votre projet

- API accessibles : Google, X, Spotify, Netflix...
- Exemple - aviation : <https://aviation-edge.com/flight-radar-and-tracker-api>
- Exemple - finance : <https://www.alphavantage.co/>
- Données en « open data » (ex. données météo, réseaux de transport, capteurs de pollution...)
- Exemples : RATP, ville de Paris  
<https://data.ratp.fr/pages/temps-reel/>  
<https://opendata.paris.fr/api/v1/console/datasets/1.0/search/>
- Outils de simulation (ex. client Python dans le lab Kinesis)