

Airflow

1. Définition du Besoin

- **Objectifs** : Identifiez les besoins principaux.
 - Exemples : Automatiser des workflows complexes, orchestrer des pipelines de données, surveiller l'exécution des tâches.
 - **Contexte** : Pourquoi Apache Airflow est-il choisi comme solution ?
 - Exemple : Besoin d'une plateforme open-source flexible et modulaire pour gérer les workflows.
 - **Contraintes** : Listez les limitations possibles (temps, ressources, expertise technique).
-

2. Estimation

- **Ressources Matérielles** :
 - Serveurs locaux ou cloud nécessaires pour exécuter Airflow.
 - Stockage requis pour les bases de données et les fichiers de logs.
 - **Temps** :
 - Découpez les tâches principales (installation, création de workflows, tests, déploiement).
 - Estimez la durée pour chaque tâche.
-

3. Documentation

- **Documentation Technique** :
 - Description de l'architecture logicielle et matérielle.
 - Guide d'installation et configuration d'Apache Airflow.
 - **Manuel Utilisateur** :
 - Guide pour exécuter et surveiller les workflows (DAGs).
 - **Processus** :
 - Notez chaque étape pour garantir la reproductibilité.
 - Documentez les dépendances Python et outils tiers utilisés.
-

4. Affectation des Tâches

- **Jawadi Marwane** :
 - Installation d'Apache Airflow et configuration initiale.
 - Création de DAGs pour des workflows spécifiques.
- **Amine Krimi** :
 - Gestion des bases de données (PostgreSQL, SQLite).
 - Intégration de tâches spécifiques dans les pipelines.

- **Collaboration :**
 - Mise en place des scripts de tests.
 - Vérification de la cohérence entre les tâches des DAGs.
-

5. Organisation des Équipes

- **Structure :**
 - Divisez les tâches en sous-équipes si nécessaire (infrastructure, développement, tests).
- **Communication :**
 - Utilisez des outils comme Teams pour coordonner les efforts.
- **Réunions :**
 - Planifiez des points réguliers pour évaluer l'avancement (daily stand-ups, revues hebdomadaires).