# Documentation Technique

## 1. Infrastructure & Déploiement

Le projet repose sur une infrastructure hybride composée d’un environnement local de développement et d’un environnement de production hébergé sur Google Cloud Platform (GCP). En local, le développeur utilise Docker Compose pour exécuter Airflow et Streamlit. En production, une machine virtuelle Compute Engine sur GCP exécute également Docker Compose, assurant la cohérence entre développement et production. Le déploiement est automatisé via GitHub Actions qui exécute les tests unitaires, se connecte à la VM via SSH et redéploie automatiquement les services.

## 2. Description des DAGs Airflow

Airflow orchestre le pipeline sous forme de DAGs. Chaque DAG comprend :  
- Collecte : API Steam → stockage brut JSON dans GCS (Bronze).  
- Transformation : nettoyage, NLP (analyse de sentiment, détection de langue).  
- Chargement : insertion des données enrichies dans Firestore (Silver/Gold).  
- Monitoring : suivi des exécutions et logs persistants.  
  
Cette organisation permet de séparer les tâches et d’assurer une reprise fine en cas d’erreur.

## 3. Gestion des secrets et sécurité

Les variables sensibles sont stockées dans un fichier .env en local, ignoré par Git. En production, elles sont gérées via GitHub Secrets. Sur GCP, les permissions sont restreintes grâce à IAM (principe du moindre privilège). Ainsi, chaque service ne dispose que des accès strictement nécessaires (ex. Firestore Writer, Storage Object Admin).

## 4. Schéma d’architecture

Le pipeline suit le cycle : API Steam → GCS (Bronze) → Transformation (Python, NLP) → Firestore (Silver/Gold) → Streamlit. Airflow orchestre le flux, GitHub Actions gère le déploiement.

## 5. Commandes utiles

- Lancer Airflow local : docker compose up airflow-webserver  
- Lancer Streamlit local : streamlit run app.py  
- Exécuter tests unitaires : pytest  
- Déployer en prod : git push origin main (déclenche workflow CI/CD)