Dossier de Maintenance

Projet : Analyse & Catégorisation des Avis Utilisateurs Amazon
Auteur : Ismaël Sylla

Sommaire

1. Introduction générale

La réussite d'un système en production ne repose pas uniquement sur sa capacité à traiter les données, mais aussi sur la manière dont il est supervisé, maintenu et amélioré dans la durée. Ce **Dossier de Maintenance** constitue le cœur opérationnel permettant d'assurer la fiabilité de la plateforme Amazon Review Analysis. Il formalise les processus, bonnes pratiques et responsabilités nécessaires pour garantir la continuité de service, gérer les incidents et planifier les évolutions.

Ce document s'inscrit dans une démarche DataOps moderne qui privilégie : - la traçabilité complète des données,
- la gestion centralisée via Airflow,
- la modularité des composants,
- la conformité aux réglementations (RGPD, AI Act),
- l'automatisation autant que possible des tâches de supervision.

L'écosystème technique repose sur une architecture distribuée comprenant Airflow, PostgreSQL en Docker, S3 en buckets segmentés, MongoDB en base NoSQL, Snowflake en entrepôt analytique et un pipeline NLP avancé. Cette diversité impose une organisation rigoureuse pour assurer une maintenance efficace.

2. Gestion des incidents

La gestion des incidents vise à stabiliser le système malgré les imprévus. Les incidents peuvent émerger d'un composant isolé (ex : chute d'une tâche Airflow) ou d'un enchaînement d'événements (ex : fichier RAW corrompu entraînant un chargement Snowflake impossible). Cette section formalise les typologies d'incidents et leurs stratégies de résolution.

2.1 Typologie des incidents

Incidents mineurs – perturbations légères

Ce sont des anomalies courantes qui ne remettent pas en cause l'exécution globale du pipeline. Elles constituent une part normale de l'exploitation et doivent être surveillées afin d'éviter qu'elles ne dégénèrent. Parmi elles : - un temps d'exécution Airflow légèrement supérieur à la moyenne,
- un volume inhabituel de logs informatifs,
- un léger retard dans le chargement d'un bucket S3.

Ces incidents nécessitent une observation mais rarement une intervention lourde.

Incidents majeurs – dégradation du service

Ils impactent la qualité du traitement ou la complétude des données : - un DAG Airflow échoue malgré plusieurs retries,
- un taux de rejets élevé apparaît dans MongoDB,
- Snowflake refuse un chargement à cause d'un schéma incompatible.

Ces incidents doivent être traités rapidement afin de préserver l'intégrité du pipeline quotidien.

Incidents critiques – interruption ou risque élevé

Ces incidents mettent à l'arrêt une ou plusieurs briques essentielles : - Airflow (scheduler ou webserver) indisponible,
- S3 inaccessible ou présentant des objets corrompus,
- Snowflake non accessible,
- MongoDB hors service,
- modèle NLP inutilisable.

Ils requièrent une intervention immédiate et coordonnée entre plusieurs équipes.

3. Processus d’escalade et responsabilités

Une chaîne d’escalade claire réduit les temps de résolution et évite l’enlisement des incidents.

Niveau 1 – Support opérationnel

Les équipes N1 se chargent : - d’effectuer les premiers contrôles (Airflow health, logs),
- d’isoler un incident reproductible,
- de vérifier les accès,
- de relancer un DAG si cela suffit.

Elles escaladent en cas de suspicion d’un problème plus profond.

Niveau 2 – Data Engineers

Ils interviennent sur : - les erreurs Airflow complexes,
- les problèmes S3 (fichiers invalides, permissions),
- les anomalies dans PostgreSQL ou MongoDB,
- les erreurs SQL ou de schéma dans Snowflake.

Ils assurent également la communication technique vers les équipes produits et support.

Niveau 3 – DevOps / Admin systèmes

Mobilisés en cas : - de panne Docker,
- de saturation des ressources serveurs,
- d’erreurs réseau ou AWS,
- d’interventions lourdes de maintenance.

4. Maintenance corrective

La maintenance corrective représente l'ensemble des actions entreprises lorsqu'un dysfonctionnement survient dans la plateforme. Elle vise à restaurer rapidement le service, à réduire l'impact sur les utilisateurs et à assurer la stabilité du pipeline en production. Dans un contexte DataOps moderne, la maintenance corrective doit être organisée, traçable et dotée de procédures claires pour garantir une résolution efficace.

4.1 Objectifs de la maintenance corrective

L'objectif principal est d'identifier rapidement l'origine d'un incident, de le résoudre et de s'assurer que le problème ne se reproduise plus. La maintenance corrective contribue directement à :

- maintenir un haut niveau de disponibilité de la plateforme,
- limiter les interruptions du pipeline Airflow,
- préserver la qualité et la fraîcheur des données,
- éviter toute propagation d'erreurs vers Snowflake ou MongoDB.

4.2 Typologie des incidents courants

La maintenance corrective couvre plusieurs catégories de problèmes, fréquents dans une architecture Data :

1. Incidents applicatifs (Airflow / scripts Python)

- Échec d'une tâche dans un DAG
- Timeout sur une extraction ou un upload S3
- Crash d'un worker Airflow
- Perte de dépendance entre deux tâches

Ces incidents se résolvent souvent via un redémarrage contrôlé de la tâche, l'analyse des logs, ou un patch correctif ponctuel.

2. Incidents d’infrastructure

- Conteneur PostgreSQL indisponible
- Stockage S3 inaccessible ou permissions IAM incorrectes
- Latence réseau entre Airflow et Snowflake
- Cluster MongoDB en surcharge

Ces incidents nécessitent une intervention plus technique et parfois une escalade vers les équipes IT ou Cloud.

3. Incidents liés à la qualité des données

- Données manquantes dans PostgreSQL
- Fichiers RAW corrompus sur S3
- Anomalies dans la transformation NLP
- Données dupliquées ou invalides

Dans ces cas, la correction passe aussi par un retraitement partiel ou complet des données concernées.

4.3 Processus standard de résolution d’incident

La maintenance corrective suit un workflow rigoureux :

1. Détection

L’incident peut être détecté via :

- les alertes automatiques d’Airflow,
- les dashboards de monitoring,
- une vérification manuelle quotidienne,

- un retour d'un utilisateur ou analyste.

2. Analyse & diagnostic

Le support analyse :

- les logs Airflow pour identifier la tâche en erreur,
- les logs S3 ou Snowflake,
- les métriques systèmes (CPU, RAM, latence réseau),
- les dernières modifications de configuration.

3. Correction

La solution peut prendre plusieurs formes :

- relance de la tâche,
- reprocessing d'une étape du pipeline,
- correction manuelle d'une donnée,
- patch temporaire du code,
- mise à jour de permissions IAM ou Snowflake.

4. Vérification

Une fois l'incident corrigé, un test systématique est réalisé :

- exécution manuelle du DAG en mode sécurisé,
- consultation des tables Snowflake et MongoDB,
- comparaison avant/après des données,
- validation via un analyste si nécessaire.

5. Documentation

Chaque incident doit être documenté :

- date, heure, contexte
- cause racine (Root Cause Analysis)
- actions entreprises
- risques associés
- actions préventives à mettre en place

Cette documentation enrichit la connaissance collective et permet d'éviter les répétitions.

5. Maintenance préventive

La maintenance préventive vise à anticiper les problèmes avant qu'ils ne surviennent. Elle s'appuie sur des routines régulières, des audits techniques et des améliorations continues de la plateforme.
Son objectif est de réduire le taux d'incidents, d'améliorer la stabilité globale du pipeline et de prolonger la durée de vie de l'infrastructure.

5.1 Principes généraux

La maintenance préventive repose sur trois piliers :

- Surveiller l'état de santé de chaque composant,
- Optimiser les traitements et la performance,
- Assainir la plateforme grâce à des opérations régulières.

Dans une architecture Airflow → S3 → Snowflake → MongoDB, la maintenance préventive est indispensable pour préserver une performance constante.

5.2 Maintenance préventive Airflow

1. Vérification de la santé du scheduler

Le scheduler Airflow doit être contrôlé régulièrement :

- absence de saturation CPU,
- absence de backlog de tâches,
- absence de worker bloqué ou zombie.

2. Nettoyage des logs

Les logs Airflow peuvent rapidement atteindre plusieurs gigaoctets.
Un nettoyage mensuel est recommandé :

- rotation automatique,
- suppression des logs de plus de 30 jours,
- archivage S3 si nécessaire.

3. Vérification des DAGs

Un audit hebdomadaire permet :

- de détecter les modifications accidentelles,
- d'identifier les tâches trop longues,
- de repérer les dépendances obsolètes.

5.3 Maintenance préventive du Data Lake S3

1. Contrôle de la croissance des fichiers RAW

Les fichiers bruts sont versionnés et accumulent rapidement du volume.
Une purge trimestrielle peut réduire le stockage inutile.

2. Analyse des accès IAM

Vérifier régulièrement :

- les permissions réellement utilisées,
- les éventuels accès non conformes,
- la rotation des clés d'accès AWS.

3. Validation de la conformité RGPD

S'assurer que :

- les fichiers ne contiennent pas de données sensibles non anonymisées,
- la durée de conservation est respectée,
- les dossiers "rejected" sont purgés automatiquement.

5.4 Maintenance préventive PostgreSQL

1. Vérification de l'espace disque

PostgreSQL peut saturer si les WAL ou snapshots ne sont pas purgés.

2. VACUUM + ANALYZE

Un nettoyage mensuel garantit :

- de meilleures performances de requêtes,
- moins de fragmentation des tables,
- une réduction du temps d'extraction.

3. Mise à jour de sécurité

Chaque montée de version doit être testée dans un environnement intermédiaire.

5.5 Maintenance préventive Snowflake

1. Suivi des coûts

Snowflake facture :

- stockage,
- compute,
- time travel.
  Un suivi hebdomadaire permet d'éviter les dérives.

2. Optimisation des vues

Certaines vues peuvent devenir coûteuses :

- optimisation des jointures,
- partitionnement,
- clustering automatique.

3. Surveillance des warehouses

Vérification :

- des périodes d'auto-suspension,
- des usages horaires,
- des pics de consommation.

5.6 Maintenance préventive MongoDB

1. Nettoyage des données anciennes

MongoDB peut s'alourdir rapidement.
Les documents obsolètes ou rejetés doivent être purgés.

2. Vérification des indexes

Un audit mensuel identifie :

- les indexes inutiles,
- les indexes manquants,
- les indexes trop volumineux.

5.7 Maintenance préventive du pipeline NLP

1. Vérification des performances du modèle

Une dérive du modèle peut apparaître avec le temps :

- variation anormale du score moyen,
- croissance des faux positifs,
- incohérences sur certaines catégories.

2. Réentraînement ou mise à jour du modèle

Recommandé tous les 6 à 12 mois.

6. Maintenance évolutive

La maintenance évolutive représente l'ensemble des actions destinées à **faire évoluer la plateforme**, à l'adapter aux nouvelles exigences métiers, aux évolutions réglementaires, aux opportunités technologiques et aux besoins des équipes Data.
Contrairement à la maintenance corrective (réagir aux incidents) ou préventive (anticiper les risques), la maintenance évolutive vise à **faire grandir le système**, étape après étape, sans interrompre la production.

Dans le contexte de l'architecture Amazon Review Analysis, cette maintenance est essentielle : les modèles NLP évoluent, les volumes de données augmentent, les catégories d'analyse changent régulièrement et les usages métiers se diversifient.
La plateforme doit donc être conçue comme un organisme vivant, capable de s'adapter continuellement.

6.1 Objectifs de la maintenance évolutive

L'objectif principal est d'assurer que la solution reste **performante, pertinente et conforme** au fil du temps. Cela implique :

- intégrer de nouvelles fonctionnalités demandées par les équipes métier,
- améliorer les performances globales du pipeline,
- ajouter des capacités d'analyse plus poussées,
- renforcer les mécanismes de sécurité,
- moderniser l'infrastructure et les modèles NLP,
- optimiser les coûts et l'ergonomie opérationnelle,
- permettre l'intégration future d'un front-end ou d'API.

La maintenance évolutive dépasse donc la simple optimisation technique : elle contribue directement à la **valeur business** de la plateforme.

6.2 Typologies d'évolutions possibles

Plusieurs axes peuvent justifier des évolutions :

1. Évolutions fonctionnelles (demandes métier)

Avec le temps, les besoins changent :

- ajout d'une nouvelle catégorie d'analyse dans le modèle NLP,
- extraction de nouveaux champs depuis PostgreSQL,
- création de vues Snowflake supplémentaires pour répondre à un besoin BI,
- intégration d'un scoring avancé (utilité de l'avis, sentiment, profondeur du commentaire),
- prise en charge de nouveaux cas utilisateurs dans MongoDB.

Ces évolutions permettent de garder un pipeline pertinent et au plus près des attentes opérationnelles.

2. Évolutions techniques

L'écosystème technologique évolue constamment :

- mise à jour des versions Python, Airflow ou Snowflake,
- optimisation des DAGs pour réduire les temps d'exécution,
- migration vers un stockage S3 plus structuré (par partition date/produit),
- introduction d'étapes de "quality gates" automatiques,
- amélioration des performances du modèle NLP via distillation, quantification ou changement d'architecture.

Ces évolutions permettent d'assurer que la plateforme reste moderne, efficace et maintenable.

3. Évolutions liées à la charge et à la scalabilité

Avec la croissance des volumes ou des usages, il devient nécessaire de :

- augmenter la capacité du warehouse Snowflake ou revoir sa stratégie d'auto-scaling,
- optimiser le partitionnement des données S3,
- scaler horizontalement Airflow en ajoutant des workers,
- placer MongoDB dans une architecture cluster pour haute disponibilité.

Ces adaptations garantissent que le pipeline reste stable même sous des charges importantes.

4. Évolutions liées à la conformité et aux régulations

La réglementation évolue chaque année (RGPD, Digital Services Act, IA Act).
Ces évolutions peuvent nécessiter :

- modification des stratégies d'anonymisation,
- renforcement des droits d'accès,
- ajout d'un mécanisme de purge automatisée,
- audit trail plus complet pour prouver la traçabilité du traitement.

La maintenance évolutive devient ici un vecteur de conformité.

6.3 Processus standard de maintenance évolutive

Une évolution doit être encadrée pour éviter les régressions et assurer la qualité de la plateforme.

Étape 1 – Recueil des besoins

Les demandes peuvent provenir :

- d'un PM,
- d'un analyste BI,
- d'un incident récurrent,
- d'une contrainte réglementaire,
- d'une nouvelle fonctionnalité imaginée par l'équipe Data.

Les besoins sont exprimés sous forme de user stories ou de fiches d'évolution.

Étape 2 – Analyse d'impact

Avant toute mise en œuvre, il faut mesurer :

- l'impact sur Airflow (nouveaux DAGs ? nouvelles dépendances ?),

- l’impact sur S3 (nouvelles partitions ? nouveaux dossiers ?),
- l’impact sur Snowflake (nouvelles tables ? nouvelles vues ?),
- l’impact sur MongoDB,
- l’impact sur les modèles NLP,
- l’impact sur la volumétrie et les coûts.

Cette analyse permet de décider si l’évolution est mineure, modérée ou majeure.

Étape 3 – Développement en environnement isolé (DEV / TEST)

L’évolution est implémentée dans un environnement séparé :

- création ou modification du DAG Airflow,
- mise à jour du code Python,
- ajout de nouveaux paramètres dans le .env,
- adaptation des modèles NLP,
- migration SQL Snowflake contrôlée.

Les tests unitaires et d’intégration doivent valider chaque changement.

Étape 4 – Tests complets (fonctionnels + performance + régression)

Les tests vérifient :

- que l’évolution fonctionne comme prévu,
- qu’elle n’a pas dégradé les performances du pipeline,
- qu’elle n’a pas introduit d'incohérence dans les données,
- que l’ensemble du pipeline Airflow reste idempotent.

Si l’évolution impacte le modèle NLP, des tests de précision, rappel et F1-score doivent être réalisés.

Étape 5 – Validation & documentation

Chaque évolution doit être :

- documentée,
- versionnée,
- intégrée à la roadmap,
- communiquée aux équipes Data / Support,
- ajoutée au changelog du système.

La documentation doit expliquer clairement le “pourquoi”, le “quoi” et le “comment”.

Étape 6 – Mise en production sécurisée

La mise en production doit respecter :

- un créneau validé (fenêtre de maintenance),
- une sauvegarde préalable (Snowflake Time Travel, dump PostgreSQL),
- un plan de retour arrière (rollback),
- une surveillance renforcée dans les heures qui suivent.

Étape 7 – Revue post-déploiement

Une fois l’évolution déployée :

- analyse des premiers résultats,
- vérification des logs Airflow,
- interrogation des tables Snowflake / MongoDB,
- consultation des utilisateurs.

Cette étape garantit que l’évolution apporte bien la valeur attendue.

6.4 Exemples concrets d’évolutions pour ton projet

1. Amélioration du modèle NLP

- intégration d'un modèle BERT multilingue,
- ajout d'un score "utilité de l'avis",
- introduction d'une détection automatique des avis frauduleux.

2. Extension du pipeline Airflow

- ajout d'un DAG "data-quality",
- automatisation de tests unitaires (Great Expectations),
- ajout d'un DAG "daily recompute" pour recalculer les indicateurs.

3. Nouvelles vues Snowflake

- vue "retour client par marque",
- vue "sentiment moyen par catégorie",
- tableau de bord "TOP 100 avis positifs".

4. Renforcement RGPD

- purge automatique trimestrielle,
- audit log complet pour tous les accès,
- configuration d'un système de pseudonymisation renforcé.

5. Mise à l'échelle

- cluster Airflow à 3 workers,
- Snowflake en warehouse Medium,
- utilisation de S3 Intelligent Tiering.

6.5 Rôle de la maintenance évolutive dans la pérennité de la plateforme

La maintenance évolutive n'est pas un simple supplément : elle est **absolument essentielle** à la longévité du projet.
Elle permet de conserver une architecture :

- performante,

- conforme,
- alignée sur les besoins métier,
- adaptée à la croissance des données,
- capable d'intégrer de nouveaux cas d'usage.

C'est grâce à elle que la plateforme Amazon Review Analysis restera pertinente et pourra s'étendre dans le futur vers :

- un front-end e-commerce enrichi,
- une API publique d'analyse des avis,
- un module de recommandations basées sur le NLP,
- un moteur d'anomalies basé sur du machine learning supervisé.

7. Matrice des risques

La mise en production d'un pipeline Data distribué sur plusieurs technologies (PostgreSQL, Airflow, S3, NLP, MongoDB, Snowflake) implique nécessairement des risques opérationnels, techniques, humains et réglementaires.
Une analyse approfondie permet d'anticiper les incidents les plus probables, d'évaluer leur impact et de définir des plans d'atténuation réalistes et activables.

L'objectif de cette matrice est double :

1. **Prévenir** les défaillances avant qu'elles n'impactent le pipeline.
2. **Outiller** les équipes pour réagir rapidement et limiter les effets domino.

7.1 Méthodologie d'évaluation

Chaque risque est évalué selon trois dimensions :

- **Probabilité (P)** : Rare / Possible / Probable
- **Impact (I)** : Faible / Modéré / Critique
- **Priorité (PxI)** : Score permettant de prioriser les actions

- **Plan d'atténuation** : Actions préventives
- **Plan de remédiation** : Actions à mettre en place en cas d'incident

Cette approche suit les grilles internes de La Poste et les bonnes pratiques AWS / Snowflake / DevOps.

7.2 Matrice détaillée des risques

🔶 **Risque 1 : Échec d'extraction PostgreSQL (connexion / authentification)**

Description :
Le pipeline Airflow dépend de PostgreSQL pour extraire les tables sources. Un mot de passe expiré, un port incorrect, un conteneur arrêté ou une saturation des connexions peut provoquer un arrêt complet du pipeline.

Probabilité : Possible
Impact : Élevé
Priorité : Haute

Causes typiques :

- Conteneur Docker PostgreSQL arrêté ou instable
- Mauvaise configuration réseau
- Trop grand nombre de connexions ouvertes (max_connections)
- Secret d'authentification expiré ou incorrect

Plan d'atténuation :

- Monitoring en continu du conteneur
- Rotation automatique des secrets
- Test de connexion automatisé dans Airflow avant l'extraction
- Limitation stricte des connexions ouvertes

Plan de remédiation :

- Redémarrage contrôlé du conteneur
- Recréation du mot de passe
- Fermeture des connexions orphelines
- Re-run de la tâche Airflow (idempotente)

🔶 Risque 2 : Erreur S3 (permissions, upload échoué, timeouts)

Description :
S3 est un élément central du pipeline. Une mauvaise permission IAM, une latence réseau ou un bucket mal configuré peut bloquer l'intégralité du flux Raw → Processed.

Probabilité : Possible
Impact : Critique
Priorité : Très haute

Causes typiques :

- Mauvais rôle IAM
- Versioning ou encryption mal configurés
- Timeouts réseau (upload > 5 sec)
- Chemins S3 incorrects

Plan d'atténuation :

- Tests réguliers d'écriture/lecture via Airflow
- IAM minimaliste et contrôlé
- Versioning toujours activé
- Validation mensuelle des policies

Plan de remédiation :

- Correction en urgence du rôle IAM
- Revalidations des chemins S3
- Relance Airflow avec backfill

🔶 Risque 3 : Défaillance Airflow (scheduler, workers, DAG bloqué)

Description :
Si le scheduler tombe ou si un DAG se retrouve en “failed” pendant plusieurs jours, la chaîne complète s’arrête, provoquant un retard critique.

Probabilité : Probable
Impact : Très élevé
Priorité : Critique

Causes possibles :

- Scheduler down
- Worker saturé ou non disponible
- Deadlock entre tasks
- Mauvaise configuration du DAG (dependency cycle)

Plan d’atténuation :

- Healthcheck Airflow activé
- Alertes Slack/Email en cas de DAG fail
- Tests pré-déploiement systématiques

Plan de remédiation :

- Redémarrage du scheduler
- Clear + rerun des tasks bloquées
- Revue manuelle du code Airflow
- Replanification du pipeline

🔶 Risque 4 : Panne de Snowflake (quota, warehouse arrêté, credentials)

Description :
Snowflake est le point d’atterrissage analytique final. Une indisponibilité du warehouse empêche le chargement des données et bloque les analyses.

Probabilité : Rare
Impact : Élevé
Priorité : Moyenne

Causes typiques :

- Warehouse auto-suspend
- Crédit Snowflake consommé
- Problème de clé d'accès
- Stage S3 inaccessible

Plan d'atténuation :

- Auto-resume activé
- Monitoring du quota de crédits
- Test quotidien du stage
- Vérification mensuelle des credentials

Plan de remédiation :

- Débloquer ou réactiver le warehouse
- Recharger les données manquantes via backfill
- Correction du stage ou de l'authentification

🔶 Risque 5 : Corruption ou mauvaise qualité des données (invalides, incomplètes, incohérentes)

Description :
Les données d'avis peuvent contenir des valeurs invalides, manquantes ou incohérentes. Sans mécanisme qualité, elles peuvent contaminer Snowflake ou MongoDB.

Probabilité : Élevée
Impact : Modéré à élevé
Priorité : Haute

Causes possibles :

- Modèle NLP qui dérive
- Valeurs manquantes ou champs invalides
- Mismatch entre les tables (join impossible)

Plan d'atténuation :

- Validation automatique dans Airflow
- Règles qualité (score, rating, date...)
- Stockage des records invalides dans rejected_reviews

Plan de remédiation :

- Analyse manuelle des lignes rejetées
- Correction en masse si nécessaire
- Retraitement des tables brutes

🔶 Risque 6 : Problème de conformité RGPD (PII mal anonymisée, logs sensibles)

Description :
Le pipeline manipule des données personnelles (PII). Un défaut d'anonymisation, un stockage non chiffré ou des logs trop détaillés expose l'équipe à un risque légal majeur.

Probabilité : Rare
Impact : Critique
Priorité : Critique

Plan d'atténuation :

- Anonymisation systématique avec hash salé
- Aucune PII dans Snowflake ou S3/Processed
- Logs purgés automatiquement (30 jours)
- Procédure de droit à l'oubli

Plan de remédiation :

- Purge immédiate des données en violation
- Notification DPO
- Correction urgente des scripts Airflow

🔶 Risque 7 : Dépendance excessive à une personne clé (Bus Factor)

Description :
Si seule une personne maîtrise Airflow, Snowflake ou l'architecture globale, la continuité d'activité devient fragile.

Probabilité : Possible
Impact : Élevé
Priorité : Moyenne

Plan d'atténuation :

- Documentation complète (ce livrable !)
- Pair-programming
- Sessions de transfert de compétences
- Standardisation des scripts

Plan de remédiation :

- Support N2
- Expertise externe si nécessaire

7.3 Synthèse visuelle des risques

Risque	Probabilité	Impact	Priorité	Niveau
Extraction PostgreSQL	Possible	Élevé	Haute	🔥
S3 – IAM/Upload	Possible	Critique	Très haute	🔥 🔥
Airflow down	Probable	Très élevé	Critique	🔥 🔥 🔥
Snowflake indispo	Rare	Élevé	Moyen	⚠️
Data Quality	Élevée	Modéré/Élevé	Haute	🔥
RGPD	Rare	Critique	Critique	🔥 🔥 🔥
Bus Factor	Possible	Élevé	Moyen	⚠️

8. Conclusion générale

Grâce à une architecture robuste, une gouvernance claire et une documentation complète, le système Amazon Review Analysis est prêt à fonctionner durablement en production. Les processus décrits garantissent une intervention rapide, une gestion transparente des incidents et une capacité d'évolution maîtrisée.