

Step 3 – Maintenance Documentation

(Version corporate harmonisée – avec schéma d'architecture, diagramme RBAC et diagramme Monitoring)
(Draft complet et prêt à livrer – procédures de maintenance, MCO/MCS, check-ups, supervision, incidents)

1. Introduction

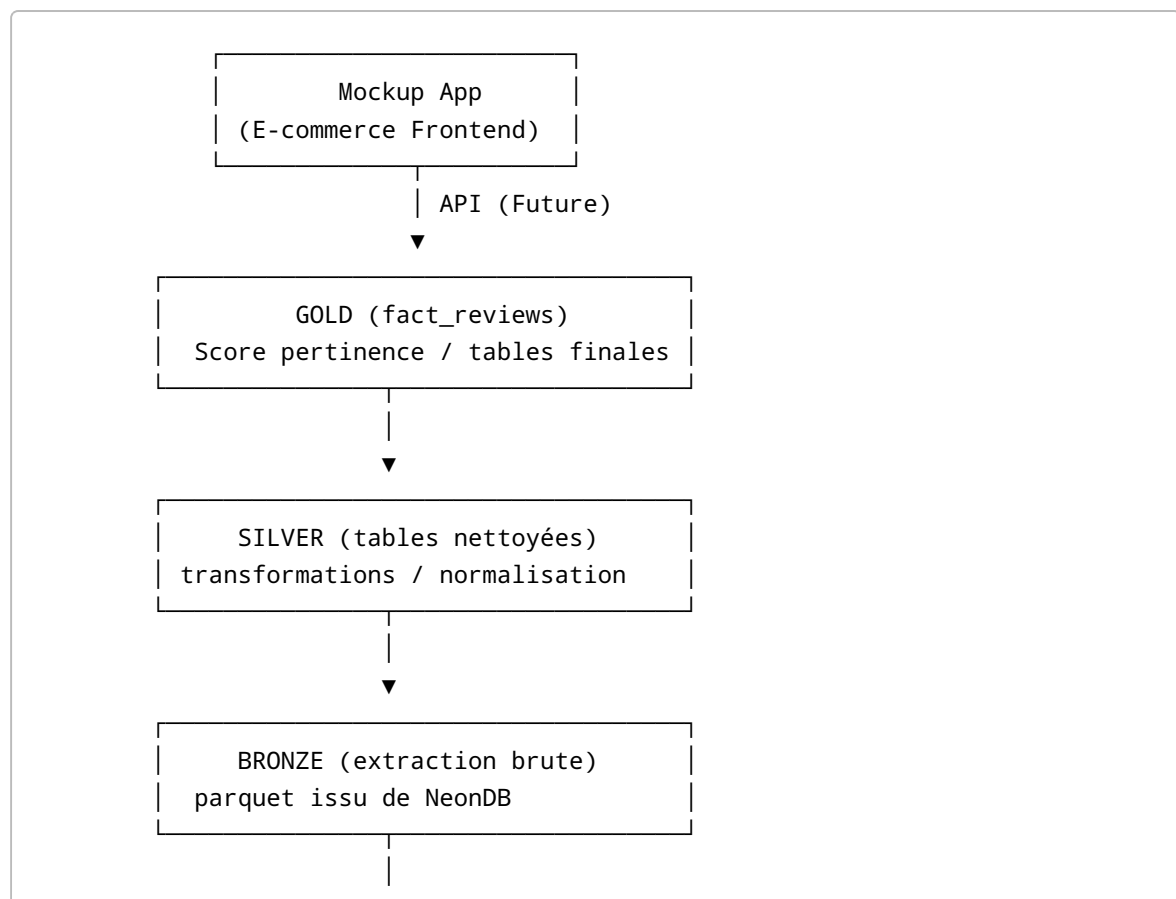
Ce document fournit l'ensemble des **procédures de maintenance opérationnelle (MCO)** et **maintenance corrective/évolutive (MCS)** pour le système de scoring et d'analyse des avis Amazon.

Il définit : - les tâches de maintenance journalière, hebdomadaire, mensuelle, - les procédures de contrôle et vérification, - les procédures de redémarrage, rollback et restauration, - les indicateurs de santé système, - la gestion des incidents et escalades, - les évolutions futures recommandées.

Cette documentation garantit la pérennité du système en production.

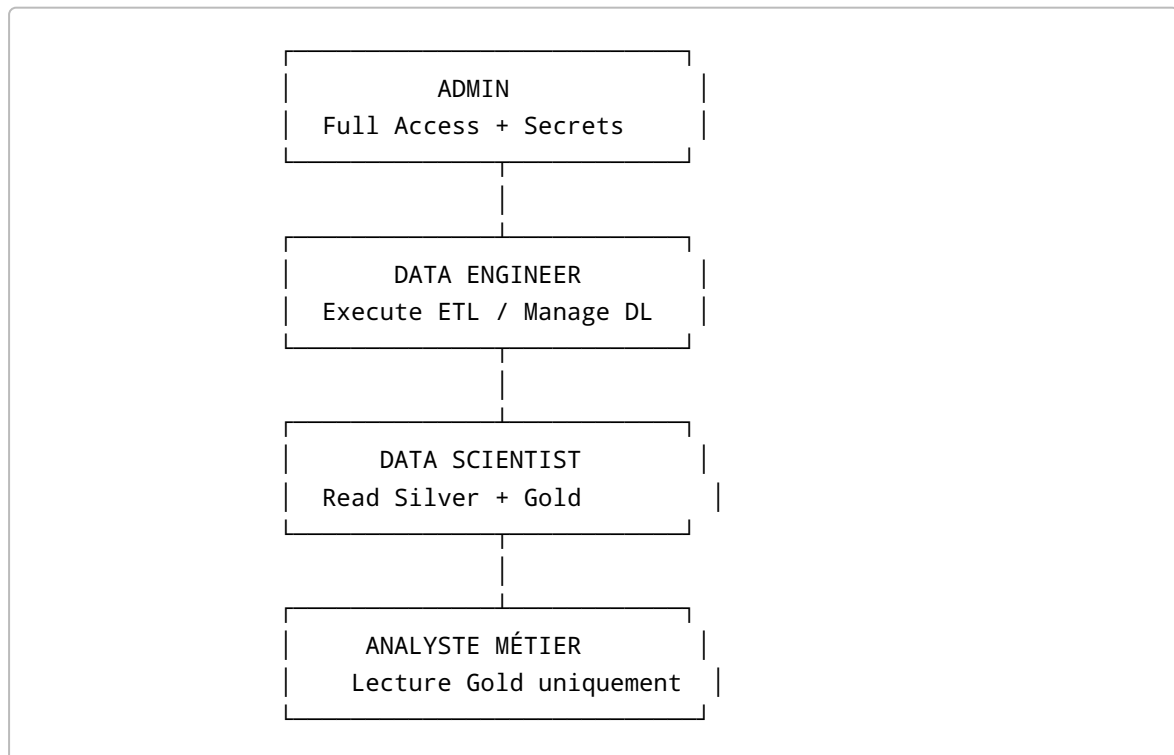
2. Architecture Maintenance Overview

2.1. Schéma d'architecture (Corporate)

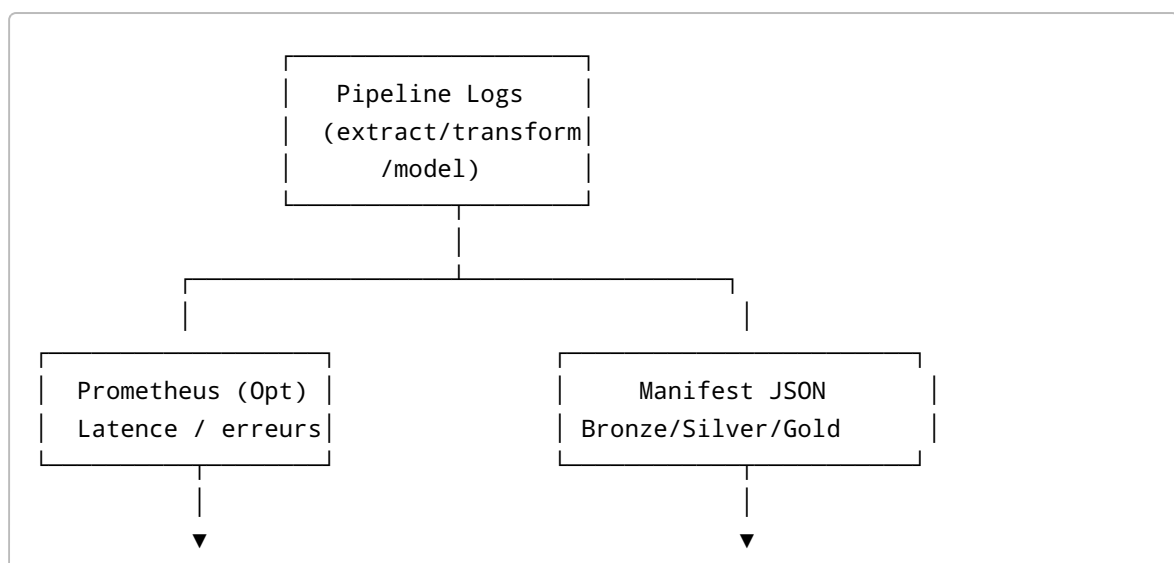


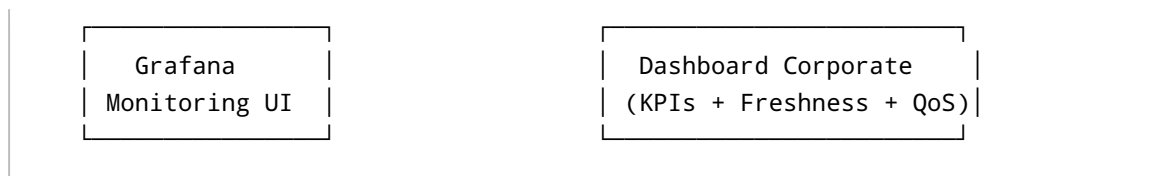


2.2. Diagramme RBAC (Corporate)



2.3. Diagramme Monitoring (Corporate)





L'architecture nécessite la supervision et la maintenance de plusieurs blocs :

Composant	Maintenance	Risques	Outils
NeonDB	intégrité DB, restauration, monitoring	corruption, montée en charge	Neon Console, PITR
ETL Pipeline	logs, latence, succès/échecs	erreurs transform / missing files	logs Python, manifest.json
Stockage Bronze/Silver/Gold	cohérence des datasets	fichiers manquants/corrompus	pandas, pyarrow
Config & Secrets	rotation des secrets, RBAC	credentials leak	.env, RBAC Neon

3. Maintenance Préventive (MCO)

3.1. Maintenance quotidienne

Action	Objectif	Comment
Vérifier les logs pipeline	détecter erreurs immédiates	<code>logs/pipeline.log</code>
Vérifier connectivité Neon	disponibilité DB	<code>SELECT 1;</code>
Vérifier génération Gold	assurer disponibilité analytique	check <code>gold/batch=...</code>

3.2. Maintenance hebdomadaire

- Vérifier les tailles des datasets (bronze/silver/gold).
- Vérifier les temps d'exécution du pipeline.
- Vérification du schéma Neon (colonne manquante, anomalies).
- Vérification de la cohérence : nombre de reviews = nombre images liés.

3.3. Maintenance mensuelle

- Test complet de restauration PITR NeoDB.
- Rotation des secrets `.env` (NEON_DB_URL).
- Archivage ou suppression des batches >90 jours.
- Audit RBAC : vérification des permissions utilisateurs.

4. Maintenance Corrective (MCS)

4.1. Procédure de redémarrage complet du système

1. Stopper exécution ETL en cours.
2. Vérifier `.env` et `config.yaml`.
3. Redémarrer pipeline :

```
python -m etl.pipeline --run-all
```

4. Vérifier bronze → silver → gold

4.2. Procédure de rollback

- Identifier dernier batch valide (gold).
- Rétablir dossier précédent :

```
cp -r gold/batch=<id_précédent> gold/batch=active
```

- Vérifier cohérence.

4.3. Procédure de réparation data lake

Symptôme : fichiers parquet manquants/corrompus.

Solution : 1. Supprimer batch Bronze/Silver 2. Relancer extract/transform

```
python -m etl.pipeline --extract  
python -m etl.pipeline --transform --batch-id=<nouveau>
```

4.4. Correction DB (Neon)

- Correction d'une table vide → relancer init CSV :

```
python -m etl.init_neon_from_csv
```

- Correction d'un schéma cassé : restauration PITR
-

5. Health Checks & Monitoring

5.1. Indicateurs système

KPI	Type	Seuil	Description
pipeline_success	disponibilité	100%	réussite ETL
pipeline_latency	performance	< 120s	temps total
db_connectivity	infra	99.9%	accessibilité DB
gold_freshness	data	<24h	génération gold

5.2. Commandes de santé

- Vérifier latence pipeline :

```
python -m etl.pipeline --extract (mesurer temps)
```

- Vérifier health DB :

```
psql "NEON_DB_URL" -c "SELECT NOW();" 
```

5.3. Monitoring externe (optionnel)

- Prometheus Exporters
- Grafana dashboards
- Status page pour pipeline success

6. Gestion des Incidents

6.1. Typologie d'incidents

Type	Exemple	Cause probable	Criticité
DB Down	Neon indisponible	maintenance / DNS	Haute
Pipeline Fail	step model missing column	erreur transformation	Haute
Data Corruption	parquet vide	extraction incomplète	Critique
Latence	ETL > 5 minutes	surcharge DB	Moyenne

6.2. Runbook Incident – Pipeline Fail

1. Lire `docs/last_run_metrics.json`

2. Trouver table fautive
3. Inspecter parquet bronze
4. Relancer transform

6.3. Runbook Incident – DB inaccessible

1. Tester connexion manuelle
2. Vérifier si Neon en incident (status page)
3. Modifier URL (pooler → direct)
4. Escalader admin

6.4. Runbook Incident – Missing gold file

1. Régénérer gold uniquement :

```
python -m etl.pipeline --model --batch-id=<id>
```

7. Évolutions Futures (Roadmap)

- Automatisation via Airflow/Kubernetes
- Mise en place d'une API temps réel pour exposer le score
- Déploiement modèle ML (sentiment analysis avancé)
- Intégration du mockup e-commerce
- Migration bronze/silver/gold vers S3

8. Conclusion

Ce document décrit l'ensemble des opérations nécessaires à la maintenance du système en production pour garantir : - sa stabilité, - sa performance, - sa résilience, - sa capacité d'évolution.

Il sert de référence aux équipes techniques pour assurer un fonctionnement optimal et durable de la solution Amazon Reviews.

(Fin du livrable Step 3 – Maintenance Documentation)