

## SENTINELLE 1

*Data Werhouse sur AWS avec le dataset Olist*

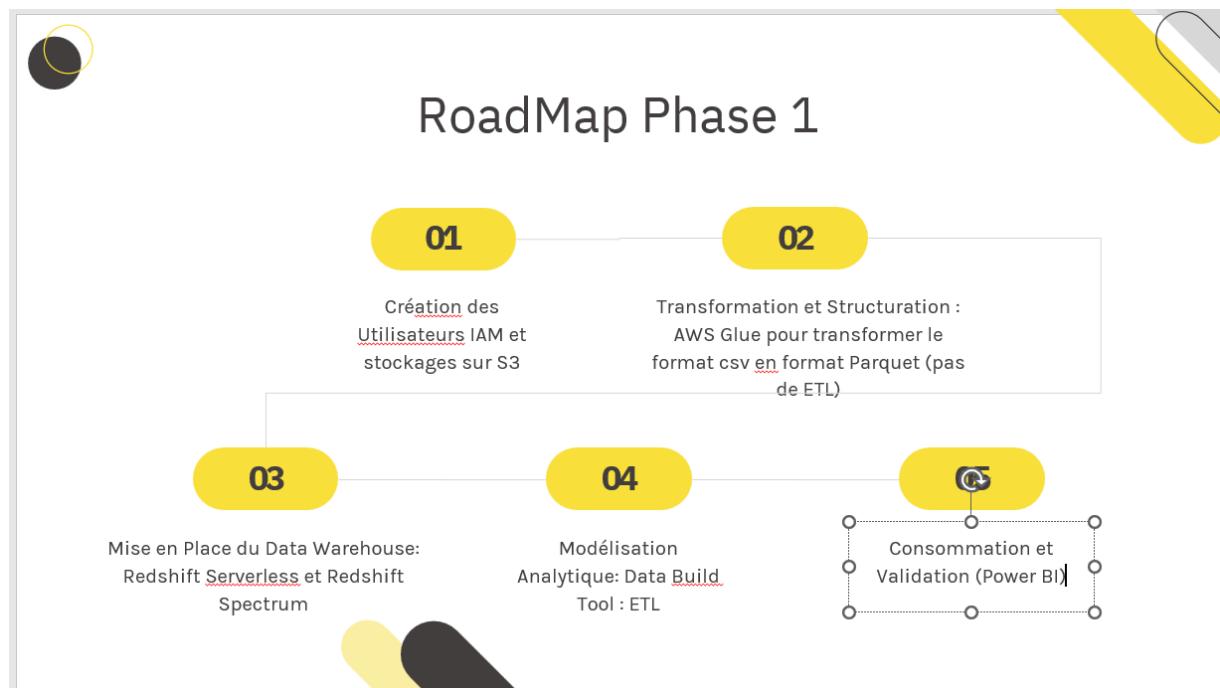
---

### *DOCUMENT TECHNIQUE : La partie Cloud*

---

Le projet sera découpé en deux phases distinctes. La première consistera à concevoir un Data Warehouse (DW) sans la partie temps réel, en utilisant uniquement les données historiques Olist. La seconde partie consistera à intégrer le flux de données en temps réel dans notre architecture, transformant ainsi le DW statique en une plateforme dynamique, un véritable Jumeau Numérique. Nous nous concentrerons immédiatement sur la Phase 1 : Data Warehouse Statique.

## Phase 1 : Data werhouse statique



Cette phase sera décomposé en 5 étapes :

### ETAPE 1: Organisation et stockages brutes

Nous commencerons par IAM (Identity and Access Management). Le rôle principal ici est le Rôle S3/Spectrum. Ce rôle doit être attaché au service Amazon Redshift et se voir accorder les permissions nécessaires pour lire les objets stockés dans notre futur bucket S3. Parallèlement, nous créerons les utilisateurs humains pour l'équipe avec les politiques d'accès minimales requises.

Ensuite, nous organiserons le Data Lake sur S3 . Nous créerons un bucket central dédié et nous le partitionnerons en zones claires : la zone bronze pour stocker les fichiers CSV historiques bruts Olist, et la zone gold qui accueillera le résultat de nos transformations modélisées (le schéma en étoile prêt pour la BI).les fichiers CSV initiaux devront être téléversés dans le dossier bronze.

## ETAPE 2 : Conversion et Structuration

Les fichiers bruts au format CSV ne sont pas optimaux pour les requêtes analytiques massives. Pour une performance et une rentabilité optimales, nous devons convertir ces données vers un format colonne tel que Parquet.

Nous utiliserons un service de traitement, **AWS Glue** (Glue Job), pour lire les 9 fichiers CSV de la zone bronze/. Le script de conversion doit effectuer la lecture et immédiatement réécrire ces données dans un nouveau chemin (bronze\_parquet/) en garantissant le format Parquet. Ce nouveau jeu de données structuré sera le point d'entrée pour Redshift Spectrum. Ce processus garantit que le moteur analytique n'aura jamais à analyser des fichiers CSV lourds, mais plutôt des fichiers Parquet compressés et performants.

## ETAPE 3 : Mise en place du Moteur Analytique (Redshift Serverless & Spectrum)

Nous avons choisi Redshift Serverless pour sa simplicité opérationnelle et son élasticité automatique, nous affranchissant de la gestion des clusters.

Le rôle S3/Spectrum créé en Étape 1 doit être attaché à cet espace de travail Redshift Serverless. Une fois Redshift opérationnel, nous allons utiliser Redshift Spectrum. Spectrum est la fonctionnalité qui nous permet de requêter directement les fichiers Parquet stockés dans S3 sans les importer dans Redshift. Pour ce faire, nous exécuterons des commandes SQL pour créer un schéma externe pointant vers le chemin bronze\_parquet/ de S3.

Amazon Redshift Serverless est un moteur de data warehouse entièrement managé qui permet d'exécuter des requêtes SQL sans avoir à gérer l'infrastructure d'un cluster. Il ajuste automatiquement la capacité de calcul en fonction des besoins et facture uniquement le temps de calcul utilisé. Cela en fait une solution simple et flexible pour analyser de gros volumes de données.

Amazon Redshift Spectrum est une extension de Redshift qui permet de lire directement des données stockées dans Amazon S3. Au lieu d'importer les fichiers dans Redshift, Spectrum interroge les données brutes ou semi-transformées là où

elles se trouvent, en exploitant des formats optimisés comme Parquet. Pour cela, on définit un schéma externe qui décrit les fichiers présents dans S3 et des tables externes qui représentent ces fichiers.

Un schéma externe agit comme une passerelle entre Redshift et S3. Il ne déplace pas les données mais établit un lien logique qui permet à Redshift Spectrum d'accéder aux fichiers Parquet stockés dans le data lake. Ce schéma contient les métadonnées nécessaires pour décrire la structure des fichiers, leurs colonnes et leurs types, ainsi que leur emplacement exact dans le répertoire bronze\_parquet/. Une fois ce schéma externe créé, des tables externes peuvent être définies. Ces tables représentent directement les fichiers Parquet présents dans S3 et peuvent être interrogées avec des requêtes SQL classiques. Elles constituent les vues brutes des données, accessibles immédiatement pour l'analyse.

L'intérêt de cette approche est multiple. Elle permet d'éviter la duplication des données et de réduire les coûts de stockage, puisque les fichiers restent dans S3. Elle améliore les performances grâce au format Parquet, qui est compressé et orienté colonnes, ce qui accélère les lectures analytiques. Elle offre également une grande flexibilité, car les données brutes ou semi-transformées peuvent être exploitées directement sans processus de chargement supplémentaire. Enfin, elle simplifie l'architecture en reliant de manière transparente le data lake et le moteur analytique Redshift Serverless.

Ainsi, le flux de la donnée est cohérent et optimisé : les fichiers bruts sont stockés dans S3, convertis en Parquet avec Glue, exposés via un schéma externe, puis interrogés avec Redshift Spectrum pour être exploités dans Redshift Serverless. Ce mécanisme garantit un pipeline efficace, économique et évolutif, parfaitement adapté à un projet de data warehouse moderne.

## ETAPE 4 : Modélisation et Préparation BI (dbt & Gold layer)

Le but de cette étape est de transformer les données brutes (les Tables Externes Spectrum) en un Schéma en Étoile optimisé pour les rapports .

Nous utiliserons dbt (Data Build Tool) pour organiser et automatiser nos transformations SQL. Dans un premier temps, dbt créera la table Fact\_Orders (Zone Silver) en joignant et nettoyant les 9 tables Olist brutes. La zone Silver joue un rôle intermédiaire : elle sert de couche de préparation où les données sont consolidées, normalisées et rendues cohérentes avant d'être utilisées pour construire le modèle final. Cette couche n'est pas toujours conservée de manière permanente. Selon les besoins du projet, elle peut être maintenue pour faciliter les audits et les contrôles de qualité, ou bien supprimée après chaque exécution si l'objectif est uniquement de produire les tables finales. Dans notre cas, la zone Silver peut être considérée

comme une étape transitoire : elle existe pour valider et fiabiliser les données, mais ce sont les tables Gold qui seront matérialisées et utilisées en production.

Ensuite, nous créerons le Gold Layer (la couche finale) qui comprendra les tables de dimensions (par exemple dim\_customers, dim\_products) et les tables de faits agrégés (par exemple fact\_sales\_agg). Ces tables Gold seront matérialisées, c'est-à-dire stockées physiquement au sein de la base de données interne de Redshift, garantissant la rapidité d'accès pour la BI et les tableaux de bord. Le Gold Layer correspond au schéma en étoile complet : les tables de faits centralisent les mesures (ventes, commandes, paiements), tandis que les tables de dimensions apportent le contexte (clients, produits, temps).

Une exécution réussie de dbt valide l'intégralité de la chaîne de transformation, depuis les données brutes exposées par Spectrum jusqu'aux tables optimisées pour la Business Intelligence. L'avantage de cette approche est multiple :

- **Clarté** : séparation nette entre les couches Bronze (brut), Silver (préparation) et Gold (final).
- **Qualité** : la zone Silver permet de détecter et corriger les incohérences avant la mise en production.
- **Performance** : les tables Gold matérialisées dans Redshift offrent des temps de réponse rapides pour les requêtes BI.
- **Traçabilité** : dbt documente automatiquement les transformations et fournit un graphe de dépendances, garantissant la transparence du pipeline.

Ainsi, la zone Silver peut être vue comme une étape de travail temporaire ou conservée selon les besoins d'audit, tandis que la zone Gold constitue la couche finale et durable, directement exploitée par les outils de reporting et d'analyse.

## ETAPE 5 : Validation par la Consommation (Power BI)

La phase se termine par la preuve que l'ensemble de l'architecture est fonctionnel et accessible aux analystes.

La connexion entre Power BI Desktop et Redshift Serverless s'effectue grâce au connecteur natif Amazon Redshift intégré dans Power BI. Ce connecteur est conçu pour établir une liaison directe avec l'endpoint Redshift Serverless. Concrètement, il faut renseigner dans Power BI l'adresse de l'endpoint, le nom de la base de données interne où les tables Gold sont matérialisées, ainsi que les identifiants IAM ou un utilisateur Redshift autorisé. Une fois ces paramètres validés, Power BI peut interroger Redshift en mode Import ou en mode DirectQuery. Le mode Import copie les données dans Power BI, tandis que le mode DirectQuery envoie les requêtes

directement vers Redshift, ce qui permet de travailler sur de gros volumes et de garder les données toujours à jour.

Nous procéderons alors à des tests. Le test clé consiste à importer les tables de la Zone Gold, en particulier la table agrégée fact\_sales\_agg, et à créer un premier tableau de bord simple, par exemple un graphique des ventes mensuelles.

L'affichage correct de ces données dans Power BI valide la chaîne de bout en bout : les fichiers bruts sont stockés dans S3, transformés en Parquet avec Glue, exposés via Redshift Spectrum, modélisés en schéma en étoile avec dbt, matérialisés dans Redshift, puis enfin visualisés dans Power BI grâce au connecteur Redshift.

Cette étape finale démontre que l'architecture est non seulement opérationnelle mais aussi exploitable pour l'analyse et la prise de décision.

## Démarche de Travail et Bonnes Pratiques

### Démarche de Travail :

Nous allons faire de L'infrastructure as code generator, ce sera du reverse engineering .

1. L'équipe suit les étapes de la Phase 1 (Data Warehouse Statique) en utilisant la Console AWS (clics). C'est la phase d'apprentissage et de validation.
2. On documente précisément les options choisies, les noms de ressources, les configurations de sécurité, et les rôles IAM utilisés. Ces informations constituent l'entrée du code CloudFormation.
3. Une fois la chaîne complète validée (Power BI accède aux données), l'architecture est considérée comme le Prototype Fonctionnel.
4. À partir de la documentation et des prototypes, l'équipe développe le code CloudFormation. Cette étape peut être accélérée grâce à l'IaC Generator d'AWS CloudFormation, qui permet de générer automatiquement des gabarits à partir des ressources créées manuellement dans la console.
5. L'architecture entière est détruite puis redéployée exclusivement via les gabarits CFN. Cela garantit la répétabilité et la conformité.

### Bonnes Pratiques :

#### 1. Organisation et Nommage (Tagging)

L'utilisation cohérente de tags est obligatoire pour la gestion des coûts, la sécurité et l'inventaire.

<b>Clé</b>	<b>Valeur (Value)</b>	<b>Explication</b>
<b>Project</b>	OlistDWSENSAE2025	Identifie toutes les ressources appartenant à ce projet.
<b>Environment</b>	Dev /Test /Prod	Nécessaire pour séparer les environnements (utile quand on passera en IaC)
<b>Owner</b>	Student-Tp2025	Responsable de la ressource pour la facturation et la gestion.
<b>CostCenter</b>	CC-Olist-statique CC-Olist-real	Pour le suivi budgétaire.

**Conseil Nommage :** Utilisez un préfixe cohérent pour toutes les ressources, ex : olist-dw-ise2025-bucket, olist-dw-ise2025-redshift, olist-dw-ise2025-glue-job, olist-dw-ise2025-role-admin-xxxxx.

## 2. Sécurité et Gestion des Identités

Les droits d'administration seront restreints aux seules ressources portant le tag du projet, garantissant ainsi un périmètre de gestion clair et sécurisé.

La sécurité est basée sur le principe du Moindre Privilège.

Tous les utilisateurs avec accès à la console AWS doivent avoir l'Authentification Multi-Facteurs activée.

Nous devons définir des accès spécifiques pour trois profils utilisateurs

- **Accès Administrateur**

Ceci concerne exclusivement les auteurs du projet et se matérialise par la création d'un IAM Group nommé olist-dw-admin.

Les politiques associées à ce groupe définissent deux niveaux de droits :

Accès Console complet : autorise la création et la modification de toutes les ressources nécessaires au projet, notamment les buckets S3, les rôles et politiques IAM, les environnements Redshift Serverless et les jobs Glue. Cet accès est

conditionné par l'utilisation du tag `Project=OlistDWSENSAE2025` , garantissant que les administrateurs ne peuvent gérer que les ressources rattachées au projet.

- Accès d'exécution: permet le lancement des Glue Jobs et l'exécution des requêtes Redshift sur les tables de la Zone Gold. Ces droits assurent la capacité de tester et valider les traitements de données tout en maintenant une gouvernance claire.

Ainsi, les administrateurs disposent d'un périmètre de gestion complet mais strictement limité au projet, ce qui garantit à la fois la flexibilité pour les auteurs et la conformité en matière de sécurité et de suivi des coûts.

- **Accès Power BI**

L'utilisateur qui consomme les données dans Power BI n'a pas besoin d'accéder à la console AWS, mais seulement à la base de données Redshift. Ici pas dessoin de créer un groupe d'utilisateur.