



## Projet EDBD | Amazon Web Service (AWS)

BASILE Francesco-Pio

REY Dorian

DURAND Elliot

Faculté des Sciences  
Université de Montpellier

2025



## Table des matières

1	Analyses des besoins métiers .....	3
1.1	Étude de cas .....	3
1.2	Nos objectifs .....	3
2	Modèles en étoiles .....	5
2.1	Facturation .....	5
2.2	Actifs .....	8
2.3	Monitoring .....	10

# 1 Analyses des besoins métiers

## 1.1 Étude de cas

Amazon Web Service (AWS) est une des filiales d'Amazon, lancée officiellement en 2006. Elle est née suite aux besoins croissant de serveurs et d'infrastructure pour le cloud. C'est aujourd'hui le leader mondial dans le domaine du cloud computing, aux côtés de Microsoft Azure et Google Cloud Platform. En 2022, 1.45 millions d'entreprises sont clients chez AWS. Ce dernier joue un rôle tellement majeur dans l'industrie que, lors de la panne majeure du 20 octobre 2025, elle aurait causé des pertes estimées à plus d'un million de dollars par heure pour les entreprises clientes, avec un impact économique total estimé à 1,4 milliard de dollars. L'entreprise possède en conséquence une infrastructure gigantesque, avec plus de 80 zones de disponibilités (une zone de disponibilité est en endroit dans lequel il y a un ou plusieurs centres de données) et une estimation de plus de 120 centres de données répartie partout dans le monde.

Le modèle économique de AWS repose sur la ventes de services. Ces derniers sont nombreux, tels que de l'hébergement, du stockage de données ainsi que des serveurs pour faire tourner des IA. La facturation est différente d'un abonnement mensuel classique, puisque les utilisateurs sont prélevés en fonction du taux d'utilisation des services ("Pay as you go"). Par exemple, pour une base de données PostgreSQL, le tarif est actuellement de 0,026\$ par Go. Chaque mois, l'utilisateur est prélevé par rapport à son utilisation global des services AWS. Cette approche permet de maximiser la rentabilité d'un service car on ne paye pas mensuellement pour un service « illimité », mais on contrôle le prix à la source du service de manière bien plus précise. Cette approche est également efficace afin de convaincre les clients de payer notre service, car ils payent seulement ce qu'ils utilisent. Ainsi, que ce soit une très grosse entreprise ou un particulier, chacun peut profiter de la meilleure technologie possible sans être bloqué par un tarif de base très élevé. Ce modèle économique est très rentable, et AWS en démontre toute la puissance. La filiale est la plus rentable de l'entreprise Amazon, avec 60% de marge net, soit 60% du chiffre d'affaire convertie en bénéfice.

## 1.2 Nos objectifs

Nos objectifs sont de vendre le plus de services possibles, tout en optimisant les coûts d'infrastructures, ces derniers étant quasiment la seule source de dépense de notre entreprise. De ce fait, les informations les plus utiles à la prise de décision au sein de l'entreprise seraient la vente de services, le taux d'utilisation ainsi que le coût réel de l'utilisation de nos services.

Nous tracerons en conséquence trois types d'informations :

- La facturation des différents services proposés, représentant le prix mensuel à payer par les clients en fonction de leur consommation. Cette information est la plus importante, l'analyse de notre unique source de revenu est indispensable. Les traitements possible de cette information seraient :
  - Le service génère le plus de chiffre d'affaire.
  - Le pays avec les utilisateurs qui paye le plus.
  - Le mois est le plus rentable de l'année.
  - Le service le plus populaire (même s'il est pas le plus rentable).
  - Le type de service par secteur d'activité et le prix moyen dépense.
- Les coûts opérationnels, représentant le coût réel d'une opération, par exemple en coût d'électricité. Cette information est la plus importante après la facturation, elle représente le cœur de l'optimisation des coûts d'infrastructure. Les traitements possible de cette information seraient :
  - La localisation qui coûte le plus cher.

- Le type de service qui pèse le plus au cours d'une année.
  - Le type de ressource le moins fiable, soit celui qui a le plus de panne.
- 
- Le monitoring, représentant le taux d'utilisation des services mais également des coût de certaines opérations dans le service. Les traitements possible de cette information seraient :
    - Le service qui consomme le plus de CPU.
    - Quelles sont les utilisateurs les plus consommateur en ressource.
    - Le mois de l'année le plus solliciter.
    - Temps réel d'utilisation d'un service.

*On peut observer les requêtes associées dans le fichier .sql situé dans le repo GitHub*

## 2 Modèles en étoiles

Pour obtenir une vue complète et multidimensionnelle de l'activité sur la plateforme AWS, nous avons développé trois Data Marts distincts. Chacun d'eux possède un modèle en étoile unique, une fonction spécifique et est optimisé pour un type d'étude ou d'analyse différent :

- Le Data Mart Facturation (Fig. 1) est le modèle principal pour l'analyse des coûts.
- Le Data Mart Actifs (Fig. 2) (si l'on considère les coûts totaux ou les états de ressources à un moment donné) est de type snapshot, car il capture une mesure à un instant  $T$  sans historique détaillé par transaction.
- Le Data Mart Monitoring (Fig. 3) est de type transactionnel, car il enregistre chaque événement ou mesure au fur et à mesure.

### 2.1 Facturation

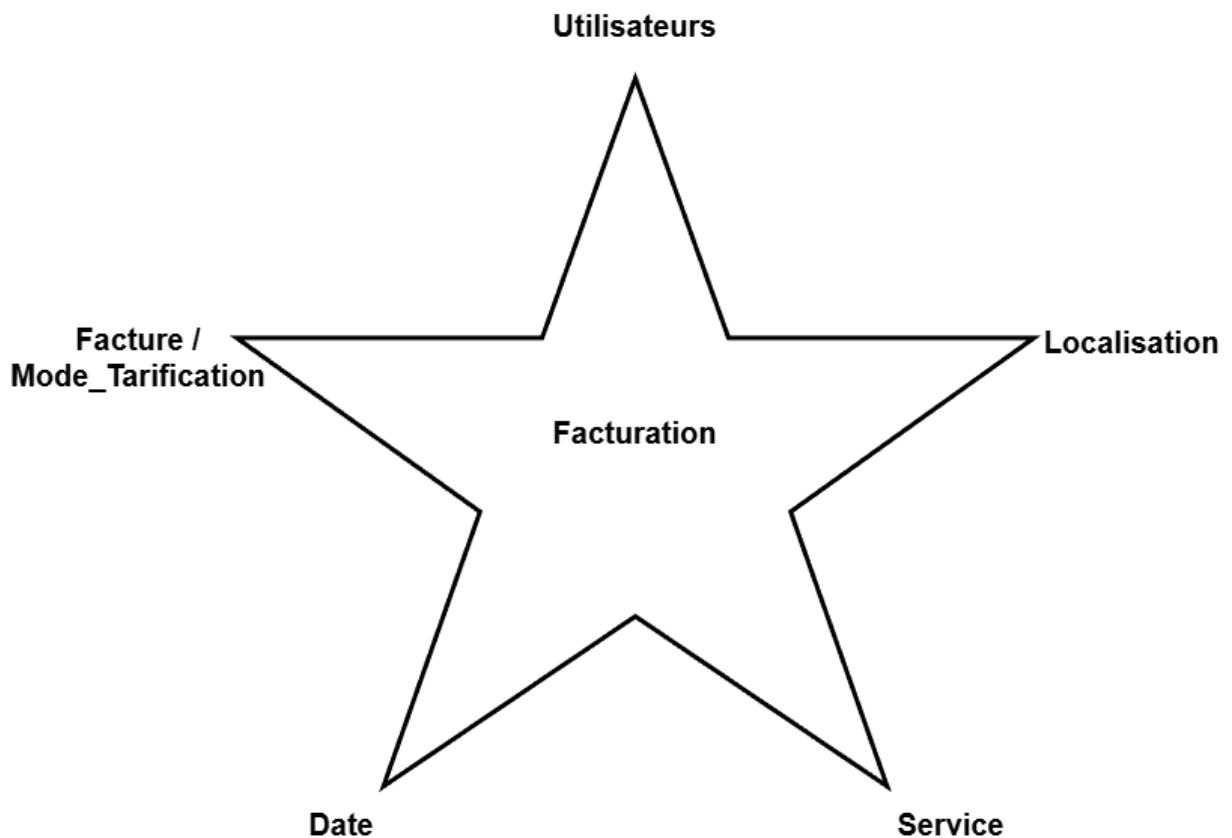


Fig. 1. – Modèle en étoile pour la facturation

<b>Utilisateurs</b>	<b>Service</b>	<b>Localisation</b>	<b>Date</b>	<b>Facturation</b>
id_utilisateur	id_service	id_localisation	id_date	total_brut
type_utilisateur	code_service	ville	date_complete	total_apres_taxes
nom_famille	nom_service	code_ville	numero_jour	total_tax
prenom	categorie	departement_province	jour	remise
sexe	famille_service	pays	numero_mois	total_apres_remise
date_de_naissance	statut_service	continent	mois	
nom_legal		fuseau_horaire	trimestre	
numero_identification		code_az	annee	
secteur_activite		nom_az	jour_semaine	
email			est_weekend	
telephone			jour_ferie	
pays_utilisateur			numero_semaine_annee	
date_inscription				

*Les clés étrangères n'apparaissent pas dans le tableau pour la visibilité mais sont bien présentes*

Nous avons deux propositions pour la dernière dimension :

<b>Facture</b>	<b>Mode_Tarification</b>
id_facture	mode_tarif_id
numero_facture	code_tarif
methode_paiement	nom_tarif
statut_facture	categorie_tarif
date_emission	type_engagement
date_echeance	duree_engagement
	option_paiement

Le modèle principal est celui de la facturation, AWS ne vendant que des services, l'entreprise ne peut pas gagner d'argent si elle n'en vend pas, ce qui reste l'objectif de base.

Les différentes mesures sont :

- total\_brut : mesure additive qui représente le total de la facture sans éléments extérieur.
- total\_apres\_tax : mesure additive qui représente le total après tax (et remise), elle varie selon le pays ou l'individu.
- total\_tax : mesure additive qui représente ce que l'utilisateur paye qui n'est pas reversé à l'entreprise.
- remise : mesure additive qui représente la réduction spécial pour l'utilisateur.
- total\_apres\_remise : mesure additive du montant après remise (hors taxe) que l'utilisateur doit payer.

On remarque que toutes les mesures sont additive, c'est car peu importe la dimension, additionner un prix aura toujours un sens.

Voici un exemple de ligne que l'on pourrait retrouver sans join dans Facturation :

- 12345, 324443, 70, 1243312, 1342523, 1000, 1100, 200, 100, 900 : le client 12345 doit/à payer 1100\$ après remise et taxe sur un prix d'origine de 1000\$ (900\$ avec remise).

Le nombre de ligne de cet entrepôt de donnée sera principalement par rapport au nombre d'utilisateur, admettons que l'entreprise a 250 000 utilisateurs actif qui au **au moins** un service, on a alors une facture minimum par mois par client donc :  $12 * 250\ 000 = 3\ 000\ 000$ .

Sachant que 250 000 est assez loin de la réalité (4.8m d'utilisateur actif en 2024) et que la limite de nombre d'entrée dans un fichier Excel (stable) est à peu près à 1.2 million, il est donc tout à fait justifié et pertinent de créer un entrepôts de données pour l'analyse de la facturation.

## 2.2 Actifs

AWS étant une plateforme qui facture l'utilisation de ses ressources, il est fondamental de pouvoir analyser ces coûts (ou « actifs »). Cette structure en étoile répond précisément à ce besoin d'analyse financière :

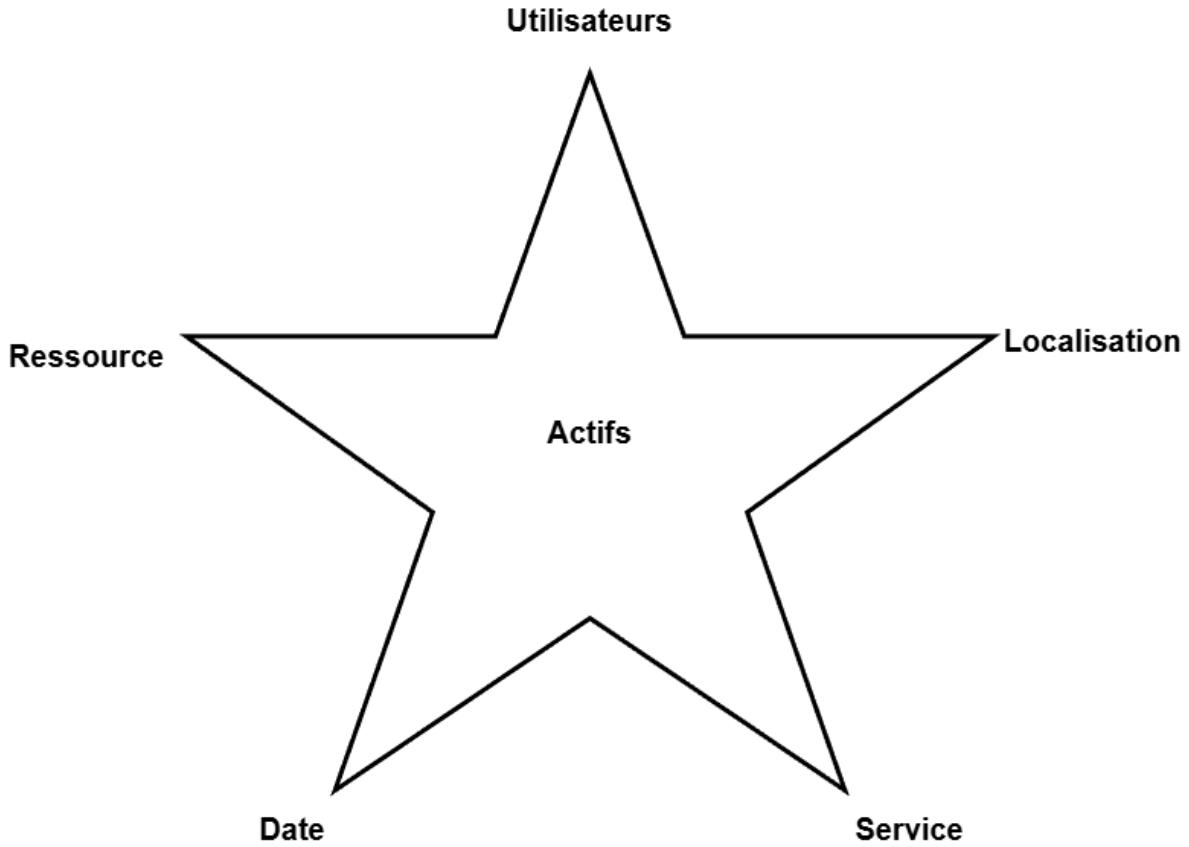


Fig. 2. – Modèle en étoile pour les actifs

Utilisateurs	Service	Localisation	Ressource	Date	Actifs
id_utilisateur	id_service	id_localisation	id_ressource	id_date	cout_total_HC
type_utilisateur	code_service	ville	nom_ressource	date_complete	cout_electricite
nom_famille	nom_service	code_ville	type_ressource	numero_jour	cout_maintenance
prenom	categorie	departement_province	statut_ressource	jour	cout_horaire_max
sex	famille_service	pays	DT_mise_en_route	numero_mois	nb_prob_materiel
date_de_naissance	statut_service	continent	DT_derniere_maintenance	mois	nb_ticket_user
nom_legal		fuseau_horaire	DT_fin_service	trimestre	
numero_identification		code_az	IP	annee	
secteur_activite		nom_az	OS	jour_semaine	
email			data_center	est_weekend	
telephone				jour_ferie	
pays_utilisateur				numero_semaine_annee	
date_inscription					

*Les clés étrangères n'apparaissent pas dans le tableau pour la visibilité mais sont bien présentes*

Notre modèle étant un snapshot, il capture les résultats agrégés ou l'état cumulé des coûts à des points temporels définis, ce qui est l'objectif de base du reporting financier. Les différentes mesures sont :

- cout\_total\_HC : mesure additive qui représente le coût total hors taxes de la ressource.
- cout\_electricite : mesure additive qui représente le coût électrique estimé de la ressource.
- cout\_maintenance : mesure additive qui représente les frais de maintenance associés à la ressource.
- cout\_horaire\_max : mesure non additive qui représente le coût horaire maximal atteint pendant la période de l'instantané.
- nb\_prob\_matériel : mesure additive qui représente le nombre d'incidents matériels survenus.
- nb\_ticket\_user : mesure additive qui représente le nombre de tickets d'assistance générés par la ressource.

On remarque que la majorité des montants sont additifs, car additionner les coûts (ex: 100 + 200) a un sens direct pour l'analyse financière. Seuls les montants maximaux ou les taux de facturation sont non additifs.

Voici un exemple de ligne que l'on pourrait retrouver sans joint dans Actifs :

- 12345, 324443, 70, 1243312, 1000, 100, 200, 100, 900 : la ressource 12345 (instance EC2) a généré un coût total hors taxe de 324443, un coût électrique de 70, et un coût de maintenance de 1243312, avec un coût horaire max de 1000, 100 problèmes matériels et 200 tickets.

Le nombre de ligne de cet entrepôt de donnée sera principalement par rapport au nombre d'instantanés (snapshots) pris par jour ou par mois, admettons que l'entreprise prend 250000 snapshots de ressources par mois, on a alors  $12 * 250000 = 3000000$  enregistrements par an au minimum, on a donc tout fait justifié de créer un entrepôts de données pour l'analyse des actifs.

## 2.3 Monitoring

AWS étant avant tout une compagnie qui loue de l'infrastructure, notamment du stockage, il est normal d'avoir une table qui « monitore » les machines louées. C'est pourquoi nous avons ce modèle en étoile :

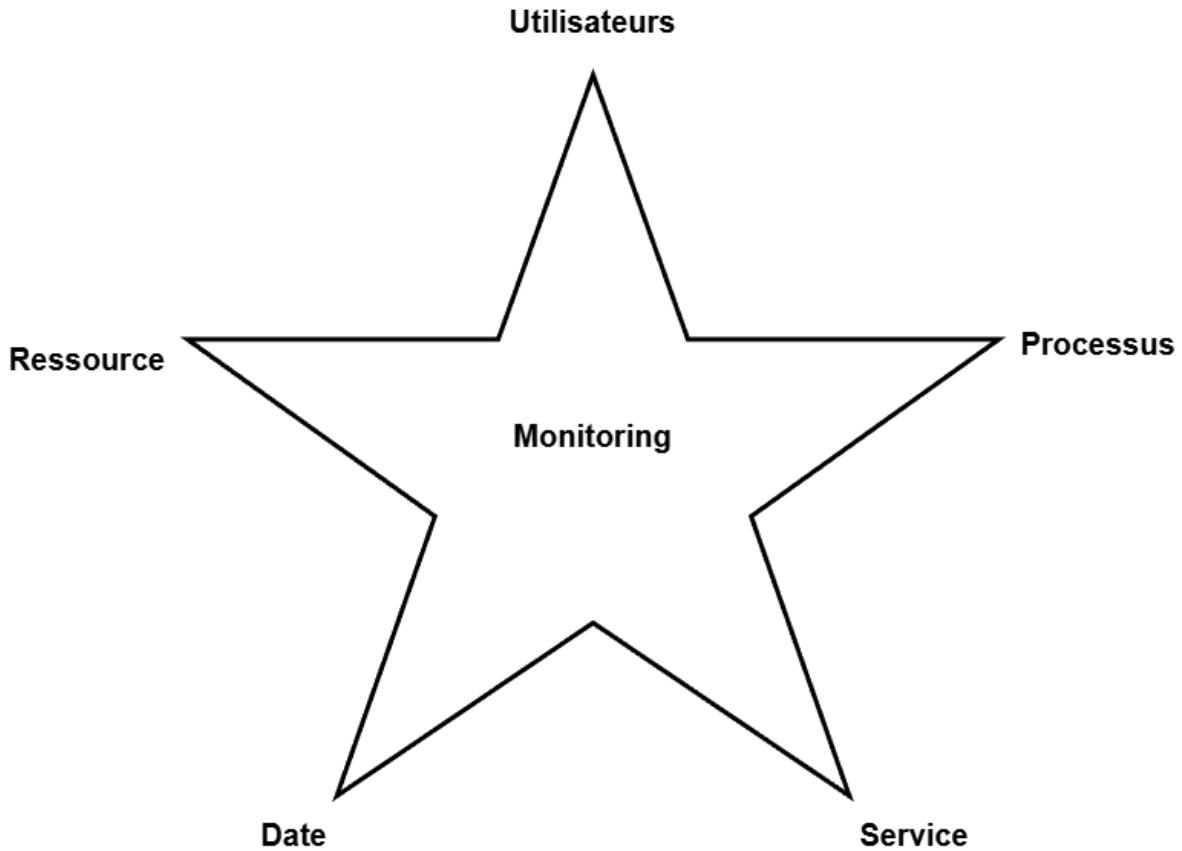


Fig. 3. – Modèle en étoile pour le monitoring

<b>Utilisateurs</b>	<b>Ressource</b>	<b>Processus</b>	<b>Service</b>	<b>Date</b>	<b>Monitoring</b>
id_utilisateur	id_ressource	id_processus	id_service	id_date	CPU_utilisation
type_utilisateur	nom_ressource	nom_processus	code_service	date_complete	GPU_utilisation
nom_famille	type_ressource	type_processus	nom_service	jour	RPM_ventilateur
prenom	status_ressource	chemin	categorie	mois	RPM_GPU_ventilateur
sexe	DT_mise_en_route	version	famille_service	trimestre	est_actif
date_de_naissance	DT_derniere_maintenance	criticite	statut_service	annee	temps_total_actif
nom_legal	DT_fin_service			timestamp	temperature
numero_identification	ip			jour_semaine	nb_core_utilises
secteur_activite	OS			est_weekend	nb_threads_utilises
email	data_center			jour_ferie	
telephone					
pays_utilisateur					
date_inscription					

*Les clés étrangères n'apparaissent pas dans le tableau pour la visibilité mais sont bien présentes*

Notre modèle étant transactionnel, il capture chaque mesure de performance sans agrégation, ce qui est l'objectif de base du suivi technique. Les différentes mesures sont :

- CPU\_utilisation : mesure non additive qui représente le taux d'utilisation du processeur.
- GPU\_utilisation : mesure non additive qui représente le taux d'utilisation de la carte graphique.
- temperature : mesure non additive qui représente la chaleur du matériel.
- temps\_total\_actif : mesure non additive qui représente le temps où la ressource est utilisée.
- nb\_core\_utilises : mesure non additive qui représente le nombre de coeurs utilisés par l'application.

On remarque que les taux d'utilisation et la température ne sont pas additives, c'est car additionner deux pourcentages ou deux températures (ex: 10% + 20%) n'a pas de sens physique. Seules les durées et les comptes peuvent être facilement additionnés.

Voici un exemple de ligne que l'on pourrait retrouver sans joint dans Monitoring :

- 12345, 12, 12, 60, 1000, 8, 4 : la ressource 12345 (par exemple l'instance EC2, qui sont les serveurs web d'AWS) affiche un taux CPU de 12% et un taux GPU de 12%, une température de 60°C, pour un temps total actif de 1000 secondes, utilisant 8 coeurs et 4 threads.

Le nombre de ligne de cet entrepôt de donnée sera principalement par rapport au nombre d'événements, admettons que l'entreprise a 250000 ressources actives qui génèrent des données toutes les 5 minutes, on a alors  $12 * 24 * 365 * 250000 = 26280000000$  enregistrements par an au minimum, on a donc tout fait justifier de créer un entrepôts de données l'analyse du monitoring.