Documentation du Projet CityFlow – Rennes Métropole

1. Contexte

Rennes Métropole fait face à une congestion routière importante sur certaines rues aux heures de pointe. La mairie et la direction des mobilités rencontrent des difficultés à anticiper les zones à aménager en priorité, à mesurer l'impact des aménagements existants et à planifier efficacement les travaux. Ce projet vise à fournir un outil d'aide à la décision basé sur l'analyse des données de trafic et de mobilité pour optimiser la gestion urbaine.

2. Problématique

Pour la mairie:

• Forte congestion sur certaines rues aux heures de pointe.
• Difficulté à anticiper les zones à aménager en priorité.
• Impact des aménagements urbains existants difficile à mesurer.
• Risque d'inefficacité dans la planification des travaux.

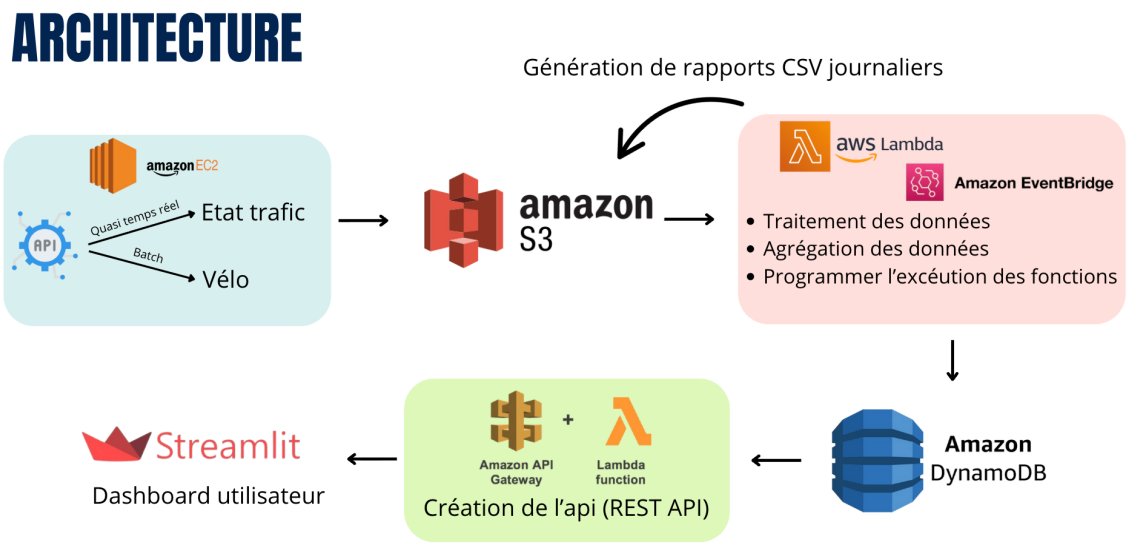
Pour la direction Mobilités et Transports :
• Identifier les lieux les plus fréquentés par les cyclistes.
• Déterminer les périodes de forte fréquentation.
• Améliorer la sécurité des cyclistes.
• Planifier les ressources et la maintenance en fonction de l'usage réel des vélos.

3. Objectifs et Solution proposée

L'objectif principal est de fournir un tableau de bord journalier permettant de suivre la congestion routière, d'identifier les zones critiques, d'analyser la fréquentation cycliste et d'optimiser la prise de décision en matière de mobilité.

Objectifs spécifiques :
• Déterminer les heures et jours où la congestion est maximale.
• Prioriser les aménagements urbains pour fluidifier le trafic.
• Évaluer l'efficacité des infrastructures existantes.
• Planifier les travaux et ajuster la circulation.
• Mettre en place des mesures temporaires.
• Optimiser le stationnement et les stations de vélos.

4. Architecture technique



5. Explication des fichiers dans git :

- Fichier Ingestion-etat-tarfic.py : Lire les données en temps reel de l'api et alimenter S3 cityflow-raw0/etat-trafic/

- Fichier ingestion_bike.py: Lire les données en batch(chaque 1h) et alimenter S3 cityflow-raw0/bike/

- Fichier lambdas/lambda-function-etat-trafic.py: traiter , aggréger et alimenter les données de l'état traffic dans DynamoDB

- Fichier lambdas/clean_bike.py : traiter et alimenter les données de vélos dans DynamoDB

- Fichier lambdas/aggregate_bike.py : traiter , aggréger et alimenter les données de vélos dans DynamoDB

- Fichier lambdas/lambda-function-rapport-etat-trafic.py : traiter , aggréger et alimenter les rapports csv relatifs à l'etat traffic dans S3

- Fichier lambdas/report_bike.py : traiter , aggréger et alimenter les rapports csv relatifs aux vélos dans S3

- Fichier lambdas/api_traffic.py : traiter une requête API : il lit, filtre et renvoie des données traffic DynamoDB en JSON avec journalisation.

- Fichier lambdas/api_vélo.py : traiter une requête API : il lit, filtre et renvoie des données vélos DynamoDB en JSON avec journalisation.

- Fichier lambdas/app_streamlit.py : contient le code pour générer le tableau de bord

6. Mode de déploiement:

L'infrastructure est hébergée **intégralement sur AWS** (Amazon Web Services).**2. Étapes du déploiement**

1. **Ingestion des données (EC2 + API Rennes Métropole)**
    - Une instance EC2 exécute les scripts Python d'ingestion en continu ou en batch.
    - Ces scripts collectent les données de trafic et de vélos, puis les stockent dans un bucket **Amazon S3**.
2. **Traitement automatique (AWS Lambda + EventBridge)**
    - **Amazon EventBridge** déclenche chaque jour une **fonction AWS Lambda**.
    - Cette Lambda lit les fichiers sur S3, agrège les données et produit un rapport CSV journalier.
    - Le code Lambda est déployé via un **package zip ou un déploiement CI/CD** (ex : GitHub Actions → AWS Lambda).
3. **Stockage & API**
    - Les données nettoyées et agrégées sont enregistrées dans **Amazon DynamoDB**.
    - Une **API REST** est exposée via **API Gateway + Lambda**, permettant de récupérer dynamiquement les données.
4. **Visualisation**
    - L'application **Streamlit** est déployée sur **Amazon EC2** et interroge l'API AWS pour afficher le tableau de bord aux utilisateurs (mairie, direction mobilités).

6. Les endpoints:

Etat trafic:

Tous les enregistrements d'un jour :
https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-trafic?date=2025-10-17

Les congestions fortes uniquement :
GET https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-trafic?niveau_congestion=Forte

Une rue spécifique :
GET https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-trafic?nom_rue=Avenue%20Fran%C3%...

Combiner les filtres :
GET https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-trafic?date=2025-11-03&nivea...

Affiche uniquement les données du département indiqué pour une date donnée.

GET https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-trafic?date=2025-10-17&depar...

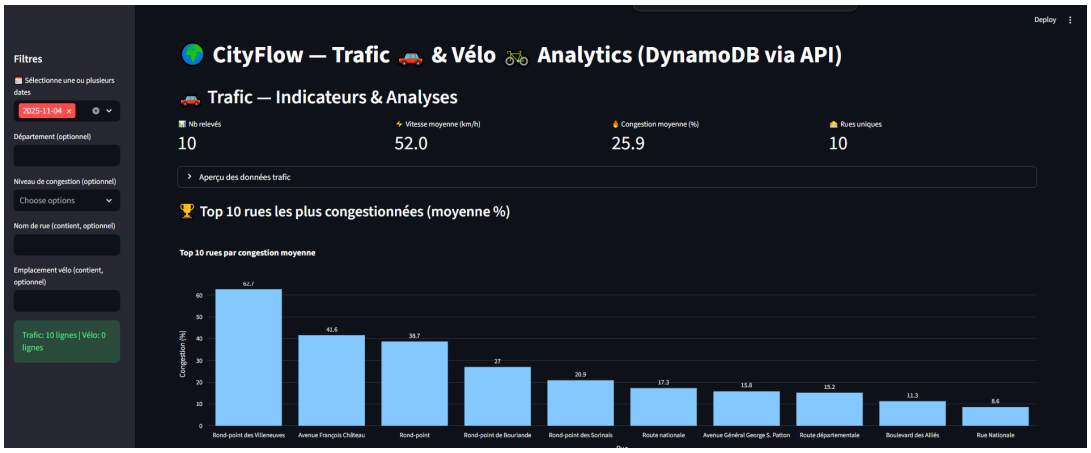
Vélos :

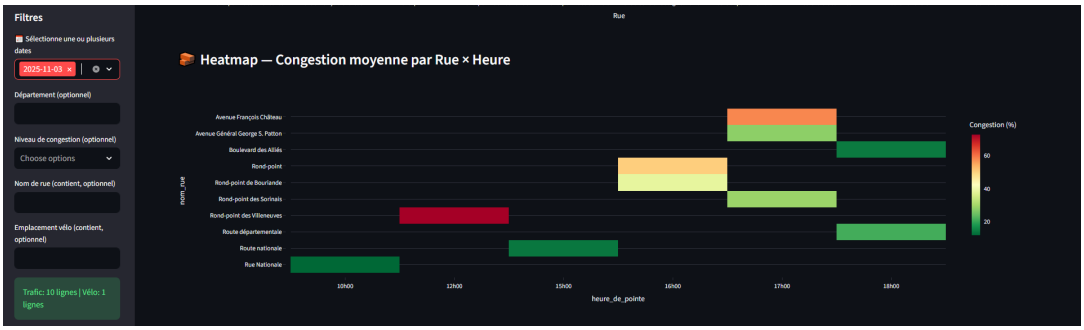
Tous les résultats
https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-velos

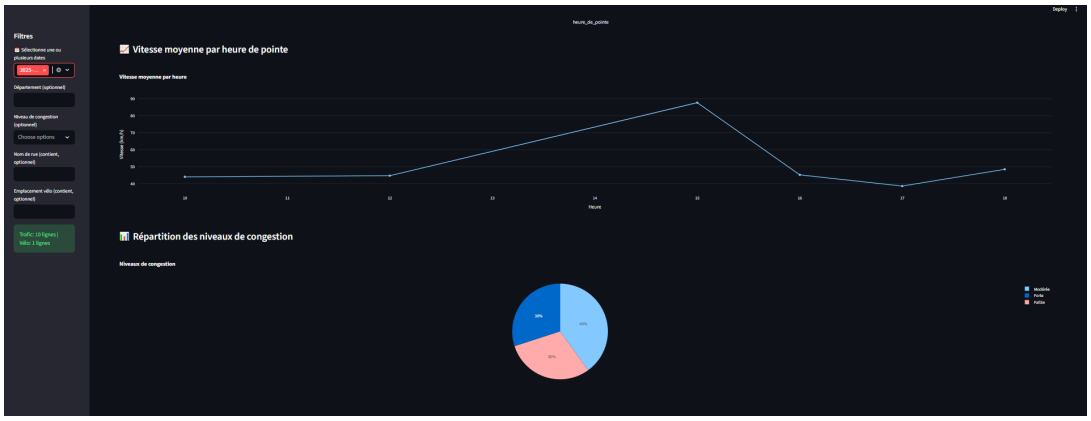
Filtrer par date
https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-velos?date=2025-09-02

Filtrer par date + lieu
https://oeagxmsmhl.execute-api.eu-west-3.amazonaws.com/stage/stats-velos?date=2025-09-02&locati...

7. L'interface streamlit




Filtres
Sélectionne une ou plusieurs dates
2025-11-03
Département (optionnel)
Niveau de congestion (optionnel)
Choose options
Nom de rue (contient, optionnel)
Emplacement vélo (contient, optionnel)
Trafic: 10 lignes | Vélo: 1 lignes
Rue
Heatmap — Congestion moyenne par Rue × Heure
Avenue François Château
Avenue Général George S. Patton
Boulevard des Alliés
Rond-point
Rond-point de Bourlande
Rond-point des Sorinais
Rond-point des Villeneuves
Route départementale
Route nationale
Rue Nationale
nom_rue
10h00
12h00
15h00
16h00
17h00
18h00
heure_de_pointe
Congestion (%)
60
40
20


Filtres
heure_de_pointe
Vitesse moyenne par heure de pointe
Vitesse moyenne par heure
Heure
Répartition des niveaux de congestion
Niveaux de congestion
Modérée
Forte
Faible


Filtres
Sélectionne une ou plusieurs dates
2025-11-03
Département (optionnel)
Niveau de congestion (optionnel)
Choose options
Nom de rue (contient, optionnel)
Emplacement vélo (contient, optionnel)
Trafic: 10 lignes | Vélo: 1 lignes
Distribution des vitesses (km/h)
Histogramme des vitesses
count
vitesse_moyenne_kmh
Vélo — Indicateurs & Tendance
Total passages (somme)
23
Moyenne (avg_counts)
4.6
Emplacements uniques
1
Aperçu des données vélo
Location_Name
Date
total_counts
avg_counts
Rennes Rue d'Isly V1
2025-11-03
23
4.6