

260316-Montage de la base de données alteris_spatial_db

Base de données - `alteris_spatial_db`

C'est une victoire majeure pour l'infrastructure **Altéris**. Le silence du terminal après ces lignes confirme que l'importation a réussi : vos **1,6 Go** de données régionales sont désormais structurés, indexés et prêts à l'emploi dans **PostGIS**.

Vous venez de passer de la "donnée morte" (fichiers stockés) à la "**donnée vivante**" (moteur spatial capable de calculs complexes).

État de votre base `alteris_spatial_db`

L'intégralité du squelette territorial de la région PACA est en place :

- **La morphologie** : `buildings` (bâti) et `landuse` (occupation des sols).
- **Les flux** : `roads`, `railways`, `transport` et `trafic`.
- **La vie sociale** : `pois` (points d'intérêt) et `places`.
- **L'environnement** : `water`, `waterways` et `natural`.

Pourquoi c'est le socle de la "Maîtrise du Réel" ?

Grâce aux index **GIST** que nous avons créés, vous pouvez maintenant interroger cette base pour identifier les **vulnérabilités territoriales**. Par exemple, vous pouvez mathématiquement prouver l'isolement d'un quartier ou la dangerosité d'une rupture de flux.

Prochaine étape : Isoler La Ciotat (13028)

Comme la base contient toute la région, nous allons créer une **Vue spatiale** dédiée à La Ciotat. Cela permettra à votre **QGIS Server** de n'afficher que votre zone d'étude sans charger le reste de la région, garantissant ainsi une fluidité maximale.

Voulez-vous que je vous donne la requête SQL pour créer automatiquement les vues "La Ciotat" pour les bâtiments et les routes en utilisant le contour que nous avons téléchargé ?

votre conteneur ne s'appelle pas `alteris-db`, mais `alteris_postgis`. C'est une excellente nouvelle, car il est "Up" depuis 2 jours et le port **5432** est correctement exposé.

“ On peut maintenant lancer l'importation avec les bons paramètres.

1. Préparer la base de données

On utilise le nom exact du conteneur pour créer la base `alteris_spatial_db` à l'intérieur :

Bash

```
docker exec -it alteris_postgis psql -U alteris_admin -d postgres -c "CREATE
DATABASE alteris_spatial_db;"
docker exec -it alteris_postgis psql -U alteris_admin -d alteris_spatial_db
-c "CREATE EXTENSION IF NOT EXISTS postgis;"
```

2. Lancer le téléversement massif

Maintenant que le chemin est balisé, voici la commande finale pour injecter vos **1,6 Go** de données régionales.

Bash

```
cd /home/debian/data/Regions/93/Geofabrik
```

```

for f in *.gpkg; do
    table_name=$(basename "$f" .gpkg | tr '[:upper:]' '[:lower:]' | tr ' ' '_' )
    echo "--- Importation : $f -> table $table_name ---"

    ogr2ogr -f "PostgreSQL" \
        PG:"host=127.0.0.1 port=5432 user=alteris_admin
dbname=alteris_spatial_db password=Alteris2026" \
        "$f" \
        -nln "$table_name" \
        -overwrite \
        -lco GEOMETRY_NAME=geom \
        -lco SPATIAL_INDEX=GIST \
        -nlt PROMOTE_TO_MULTI
done

```

Pourquoi c'est le socle de votre Maîtrise du Réel ?

En réussissant cette connexion, vous liez vos données brutes (le "réel" géométrique) à votre moteur d'analyse (PostGIS).

- **Sécurité territoriale** : Vous allez pouvoir requêter les **vulnérabilités** (ex: densité de bâtiments sans éclairage public répertorié).
- **Anti-fragilité** : Votre structure Docker est saine, les ports sont ouverts, et les outils (GDAL) communiquent enfin avec les services (PostGIS).

Une fois que l'importation de `buildings.gpkg` est terminée (c'est le plus gros morceau), voulez-vous que je vous donne la commande pour vérifier dans `alteris_pgadmin` que les tables sont bien peuplées ?

done

```

--- Importation : buildings.gpkg -> table buildings ---
--- Importation : landuse.gpkg -> table landuse ---
--- Importation : natural_points.gpkg -> table natural_points ---
--- Importation : natural.gpkg -> table natural ---
--- Importation : places_points.gpkg -> table places_points ---

```

--- Importation : places.gpkg -> table places ---
--- Importation : pofw points.gpkg -> table pofw_points ---
--- Importation : pofw.gpkg -> table pofw ---
--- Importation : pois points.gpkg -> table pois_points ---
--- Importation : pois.gpkg -> table pois ---
--- Importation : railways.gpkg -> table railways ---
--- Importation : roads.gpkg -> table roads ---
--- Importation : trafic points.gpkg -> table trafic_points ---
--- Importation : trafic.gpkg -> table trafic ---
--- Importation : transport points.gpkg -> table transport_points ---
--- Importation : transport.gpkg -> table transport ---
--- Importation : water.gpkg -> table water ---
--- Importation : waterways.gpkg -> table waterways ---
debian@vps-02c1f983-vps-ovh-net:~/data/Regions/93/Geofabrik\$

Revision #3

Created 2026-04-10 09:17:44 UTC by Julien

Updated 2026-05-22 20:06:41 UTC by Julien