

Votre première carte OSM locale sur Ubuntu 24.04

Suivez cette feuille de route pour mettre en place une pile de tuiles OpenStreetMap fonctionnelle, depuis Ubuntu 24.04 jusqu'à l'affichage d'une première tuile. Commencez par un extrait régional afin de valider chaque maillon avant d'envisager une zone plus vaste.

Au sommaire

- 1 Objectif et périmètre
- 2 Avant de commencer
- 3 Installer et valider la pile cartographique
- 4 Repères de configuration à ne pas perdre
- 5 Préparer les données et publier les premières tuiles
- 6 Validation finale

Objectif et périmètre

Ce parcours cible un serveur Ubuntu 24.04 LTS amd64 et un extrait régional OpenStreetMap au format .osm.pbf. PostgreSQL et PostGIS stockent les données, osm2pgsql les importe, Mapnik les dessine, renderd prépare le rendu et Apache avec mod_tile distribue les tuiles au navigateur.

 Fichier PBF

 Import osm2pgsql

 Base PostGIS

 Style Mapnik

 Service renderd

 Apache mod_tile

 Tuiles navigateur

Avant de commencer

Vérifier que le système est Ubuntu 24.04 LTS amd64

Contrôle : `lsb_release -a`

Disposer d'un compte administrateur avec sudo

Contrôle : `sudo -v`

Contrôler la mémoire disponible

Repère de départ pour un petit extrait : 8 à 16 Go, selon la documentation Switch2OSM

Contrôler l'espace disque

Prévoir au moins 100 Go comme point de départ pour un petit extrait, puis confirmer selon la zone

Vérifier que la locale UTF-8 est active

Contrôle : `locale`

Préparer un extrait régional cohérent

Télécharger un fichier .osm.pbf depuis Geofabrik plutôt que les données mondiales

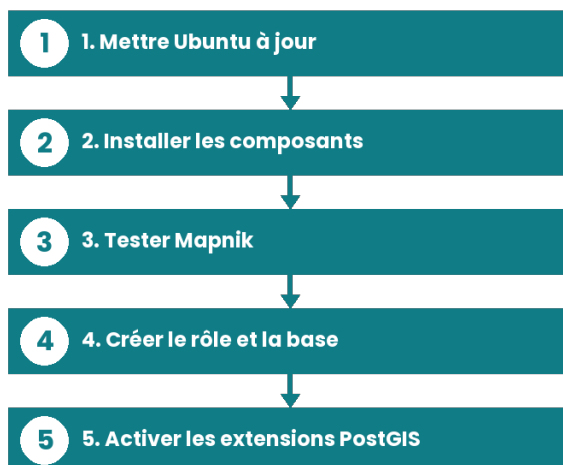
Prévoir un instantané ou une machine réversible

L'import et la configuration modifient durablement le serveur

Conserver le nom du compte Linux utilisé

Il sera repris dans le chemin du style Mapnik

Installer et valider la pile cartographique



1. Mettre Ubuntu à jour

Depuis une session SSH, exécutez : `sudo apt update` puis `sudo apt upgrade -y`. Vérifiez ensuite la version avec `lsb_release -a`. Si un nouveau noyau a été installé, redémarrez avant de continuer.

2. Installer les composants

Installez PostgreSQL, PostGIS, `osm2pgsql`, Mapnik, `renderd`, `mod_tile`, Apache et leurs dépendances avec la commande de l'article. Contrôlez ensuite les paquets avec : `dpkg -l | grep -E 'postgresql|postgis|osm2pgsql|mapnik|mod_tile|renderd'`. Plusieurs lignes doivent commencer par `ii`.

3. Tester Mapnik

Lancez : `python3 -c 'import mapnik'`. La commande doit revenir sans erreur. En cas de `ModuleNotFoundError`, réinstallez `mapnik-utils` et `python3-mapnik`, puis refaites le test.

4. Créer le rôle et la base

Créez le rôle PostgreSQL `_renderd`, puis la base UTF-8 gis qui lui appartient : `sudo -u postgres -i createuser _renderd` et `createdb -E UTF8 -O _renderd gis`.

5. Activer les extensions PostGIS

Dans la base gis, activez les extensions `postgis` et `hstore`, puis attribuez `geometry_columns` et `spatial_ref_sys` au rôle `_renderd`. Vérifiez le résultat avec : `sudo -u postgres psql -d gis -c '\dx'`.

Repères de configuration à ne pas perdre

Le nom de la base est `gis` et doit rester cohérent dans l'import, le style et la configuration du rendu.

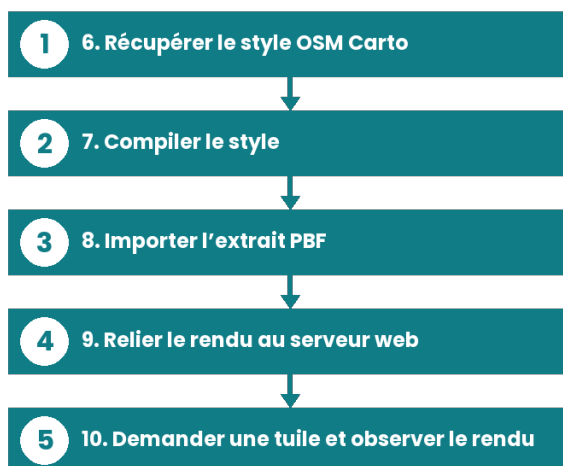
La pile décrite utilise PostgreSQL 16, PostGIS 3, `osm2pgsql`, Mapnik, `renderd` et Apache.

La source de données est un extrait régional `.osm.pbf`, pas la planète OpenStreetMap complète.

Le rôle PostgreSQL `_renderd` doit pouvoir accéder à la base sans utiliser le compte administrateur PostgreSQL.

Le style utilisé est `openstreetmap-carto v5.9.0`, compilé avec Carto 1.2.0 pour produire `mapnik.xml`.

Préparer les données et publier les premières tuiles



1 6. Récupérer le style OSM Carto

Dans un répertoire de sources de votre dossier personnel, clonez `openstreetmap-carto` puis positionnez-vous explicitement sur la version `v5.9.0`. Respectez le chemin associé à votre compte Linux.

2 7. Compiler le style

Utilisez Carto 1.2.0 pour compiler le fichier `project.mml` et produire `mapnik.xml`. Ce fichier devient la configuration de style consommée par Mapnik et `renderd`.

3 8. Importer l'extrait PBF

Importez le fichier `.osm.pbf` régional dans la base gis avec `osm2pgsql`, en utilisant les fichiers Lua et de style du projet `openstreetmap-carto`. Le style doit être préparé avant l'import.

4 **9. Relier le rendu au serveur web**

Configurez renderd avec le style compilé, puis Apache et mod_tile pour distribuer les images de tuiles. PostgreSQL ne doit pas être exposé directement sur Internet : Apache est la couche web à publier.

5 **10. Demander une tuile et observer le rendu**

Ouvrez dans un navigateur l'URL de tuile configurée par Apache et mod_tile. Pendant la requête, surveillez les journaux de renderd afin de vérifier que la demande arrive bien jusqu'au moteur de rendu.

! Une carte vide n'est pas un diagnostic

Si aucune carte ne s'affiche, vérifiez successivement la présence de renderd et de libapache2-mod-tile, l'accès du rôle _renderd à la base gis, la présence de postgis et hstore, la validité de mapnik.xml et les journaux de renderd pendant une requête de tuile. N'élargissez pas la zone d'import avant d'avoir validé un extrait régional.

Validation finale

Votre première mise en service est validée lorsque l'extrait est présent dans gis, que le style openstreetmap-carto est compilé, que renderd démarre sans erreur observable et qu'une requête passant par Apache et mod_tile déclenche effectivement le rendu d'une tuile. Mesurez ensuite l'espace disque, la mémoire et le temps d'import avant tout changement d'échelle.

Les besoins en mémoire et en disque sont des repères de départ issus de la documentation Switch2OSM citée dans l'article ; ils dépendent de la zone importée, du cache et du trafic. Cette procédure ne constitue pas une recommandation pour un import mondial ou un serveur de production.