

Votre premier serveur de tuiles OSM, du PBF à la carte

Suivez cette feuille de route pour préparer Ubuntu 24.04, importer un extrait régional OpenStreetMap et vérifier que votre serveur produit bien ses premières tuiles. Cochez chaque contrôle avant de passer à l'étape suivante.

Au sommaire

- 1 Avant de commencer
- 2 1. Préparer Ubuntu et les outils
- 3 2. Installer la base géographique
- 4 3. Importer les données OSM
- 5 4. Relier les composants et valider le rendu
- 6 Contrôle final avant publication

Avant de commencer

- ☐ **Vérifier qu'Ubuntu 24.04 LTS amd64 est installé**
Contrôle : `lsb_release -a`
- ☐ **Disposer d'un compte avec des droits sudo**
Contrôle : `sudo -v`
- ☐ **Prévoir au moins 16 Go de RAM**
Contrôle : `free -h` ; 24 Go peuvent réduire les risques pendant l'import
- ☐ **Prévoir au moins 50 Go d'espace libre**
Contrôle : `df -h /`
- ☐ **Vérifier l'accès réseau**
Il sera nécessaire pour les paquets, le style et l'extrait OSM
- ☐ **Choisir un extrait régional au format .osm.pbf**
Éviter de commencer par l'import de la planète entière
- ☐ **Créer un instantané ou une sauvegarde de la machine**
À effectuer avant le premier import



Fichier PBF



Import osm2pgsql



PostgreSQL PostGIS



Rendu Mapnik



Service renderd

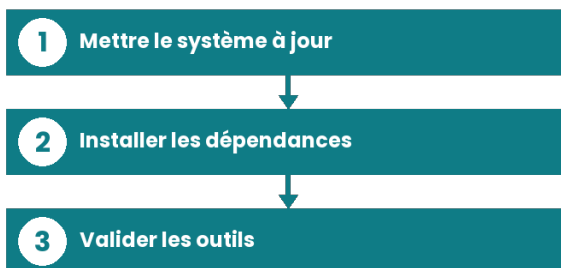


mod_tile Apache



Tuiles navigateur

1. Préparer Ubuntu et les outils



1 Mettre le système à jour

Exécutez : `sudo apt update && sudo apt upgrade -y`. Contrôlez ensuite l'environnement avec `lsb_release -a`, `apt list --upgradable` et `locale`.

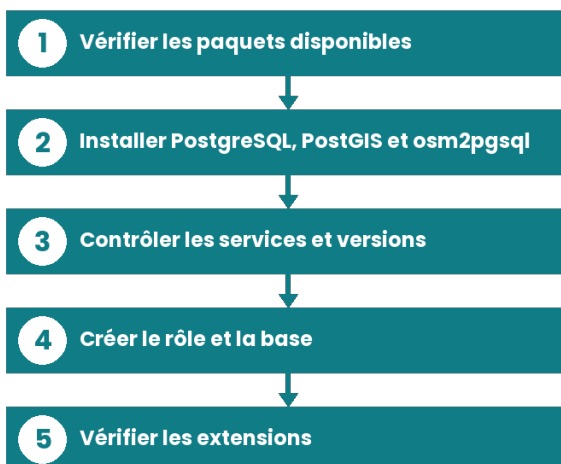
2 Installer les dépendances

Installez les outils système, géographiques et de compilation nécessaires avec : `sudo apt install -y ca-certificates gnupg curl wget unzip bzip2 tar build-essential git python3 python3-distutils gdal-bin mapnik-utils nodejs npm`

3 Valider les outils

Avant de poursuivre, vérifiez la présence de Git, Python, Node.js et npm avec `git --version`, `python3 --version`, `node --version` et `npm --version`.

2. Installer la base géographique



1 Vérifier les paquets disponibles

Contrôlez les versions proposées par vos dépôts Ubuntu avec : `apt-cache policy postgresql-15 postgresql-15-postgis-3 osm2pgsql`. Si les paquets attendus ne sont pas disponibles, arrêtez-vous avant d'ajouter un dépôt tiers.

2 Installer PostgreSQL, PostGIS et osm2pgsql

Installez : `sudo apt install -y postgresql-15 postgis postgresql-15-postgis-3 osm2pgsql`

3 Contrôler les services et versions

Utilisez `psql --version`, `osm2pgsql --version`, `sudo systemctl is-active postgresql` et `sudo -u postgres psql -c "SELECT version();"` . PostgreSQL doit rester accessible localement.

4 Créer le rôle et la base

Créez le rôle `_renderd`, la base `gis` et les extensions spatiales : `sudo -u postgres createuser _renderd ; sudo -u postgres createdb -E UTF8 -O _renderd gis ;` puis activez PostGIS et `hstore` dans la base `gis`.

5 Vérifier les extensions

Contrôlez leur présence avec : `sudo -u postgres psql -d gis -c "\dx"`. La liste doit inclure PostGIS et `hstore`.

! Ne publiez pas PostgreSQL

La base de données doit rester sur l'interface locale. Le composant web à publier est Apache ; PostgreSQL n'a pas besoin d'être exposé sur Internet pour servir les tuiles.

3. Importer les données OSM

1 Télécharger un extrait régional

2 Préparer l'import

3 Surveiller les ressources

1 Télécharger un extrait régional

Créez un répertoire de données situé hors du dossier web, puis récupérez un fichier régional `.osm.pbf` depuis une source d'extraits comme Geofabrik. Remplacez l'exemple régional du guide par la zone réellement nécessaire.

2 Préparer l'import

L'outil `osm2pgsql` transforme le fichier PBF en tables exploitables par Mapnik dans PostgreSQL. Vérifiez l'espace disque, la mémoire disponible et les droits du compte `_renderd` avant de lancer l'opération.

3 Surveiller les ressources

Un import peut échouer par manque de mémoire, d'espace libre ou à cause de droits incohérents. La taille du PBF, les index PostGIS et le cache influencent les besoins réels.

4. Relier les composants et valider le rendu

Générez le style cartographique Mapnik à partir des ressources OSM Carto.

Reliez `renderd` et `mod_tile` à Apache afin de publier les tuiles.

Utilisez le compte système `_renderd` de manière cohérente pour la base, le rendu et les fichiers de cache.

Demandez une tuile locale au format PNG avant de tester l'affichage dans un navigateur.

Le contrôle décisif est une réponse HTTP 200 sur une URL de tuile.

Affichez ensuite la carte dans une page Leaflet pour confirmer le fonctionnement de bout en bout.

Contrôle final avant publication

- ☐ **La base gis contient PostGIS et hstore**
Vérifier avec la liste des extensions PostgreSQL
- ☐ **L'extrait régional a été importé par osm2pgsql**
Confirmer que les données sont disponibles pour le rendu
- ☐ **Le style Mapnik est généré**
Vérifier les dépendances de style avant la configuration du rendu
- ☐ **renderd et mod_tile sont reliés à Apache**
Apache est le point d'entrée web
- ☐ **Une tuile PNG répond en HTTP 200**
Effectuer ce test local avant tout diagnostic côté navigateur
- ☐ **La carte s'affiche dans Leaflet**
Dernière vérification fonctionnelle
- ☐ **PostgreSQL n'est pas exposé publiquement**
Conserver l'accès de la base sur l'interface locale

Les repères matériels indiqués sont des minimums de préparation et ne garantissent pas une capacité donnée : les besoins réels dépendent notamment de la taille du fichier PBF, des index, du cache et de la fréquentation.