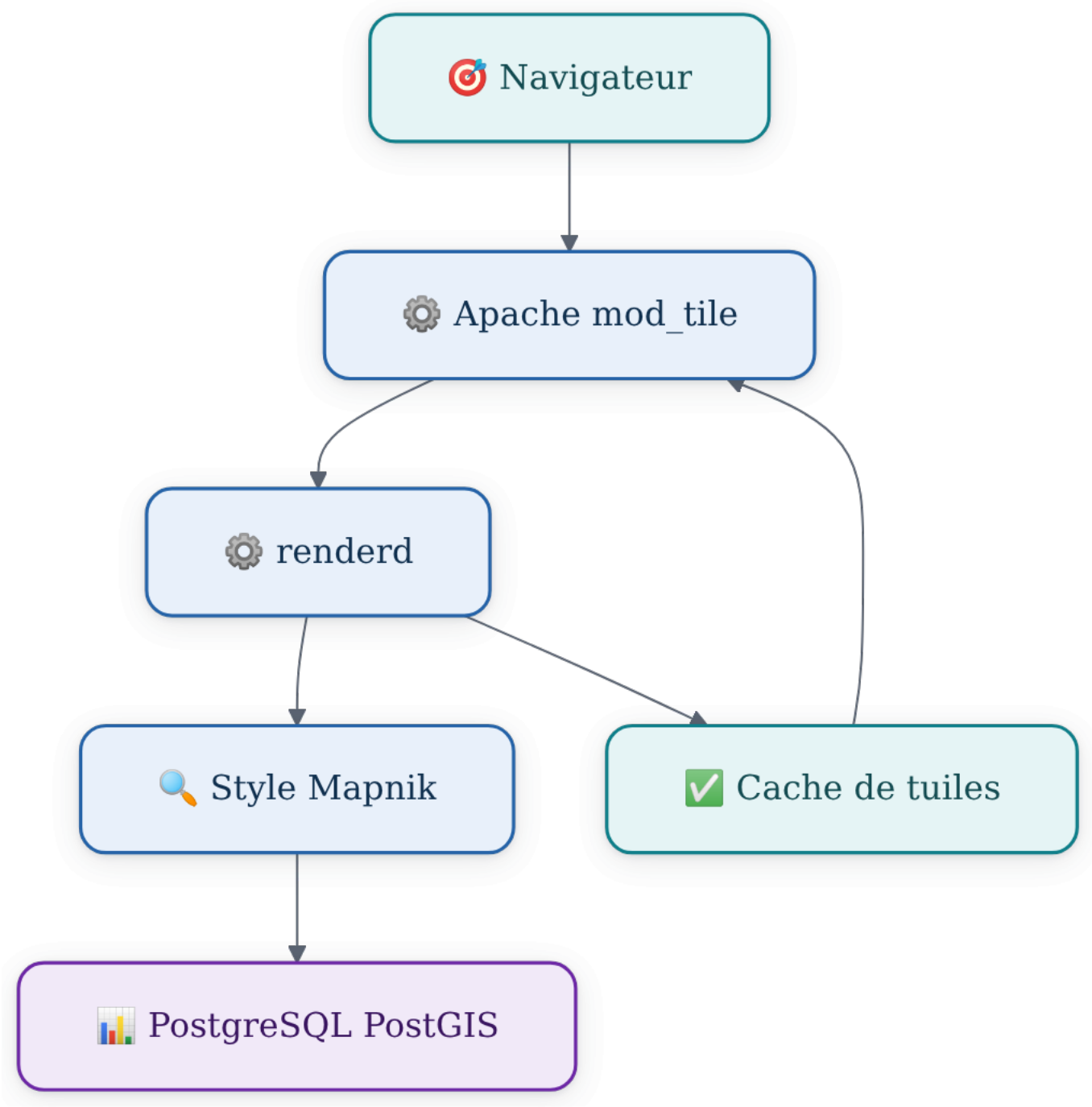
Votre premier serveur de tuiles OSM, de zéro à la carte

Suivez cette feuille de route pour préparer Ubuntu 24.04, importer un extrait OSM et mettre en place la pile de rendu. Commencez par une zone régionale afin de valider chaque maillon avant d'envisager une couverture plus large.

Au sommaire

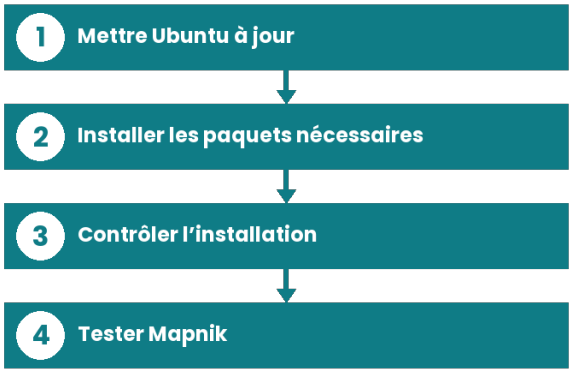
Avant de commencer

- [ ] **Vérifier que le système est Ubuntu 24.04 LTS**
  Commande : lsb_release -a
- [ ] **Contrôler la mémoire disponible**
  Commande : free -h ; l'import dépend de la taille de l'extrait.
- [ ] **Contrôler l'espace disque**
  Commande : df -h ; prévoyez au moins 18 Go pour un petit extrait.
- [ ] **Vérifier les droits sudo**
  Commande : sudo -v
- [ ] **Choisir un extrait régional au format .osm.pbf**
  Adaptez la zone à ce que votre application doit réellement afficher.
- [ ] **Prévoir une sauvegarde avant tout nouvel import destructif**
  L'option --create reconstruit les tables OSM de la base cible.



INFOGRAPHIE

1. Installer les composants de la pile



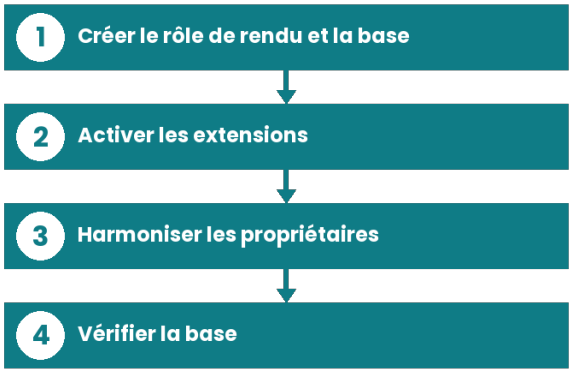
1 **Mettre Ubuntu à jour**
Exécutez sudo apt update puis sudo apt upgrade. Une mise à niveau peut nécessiter un redémarrage.

2 **Installer les paquets nécessaires**
Installez PostgreSQL, PostGIS, osm2pgsql, Mapnik, renderd, mod_tile, Apache et les dépendances du style avec les paquets natifs d'Ubuntu 24.04. N'ajoutez pas de PPA obsolète.

3 **Contrôler l'installation**
Utilisez dpkg -l | grep -E 'postgresql|postgis|osm2pgsql|mapnik|mod_tile|renderd'. Chaque composant attendu doit apparaître avant de poursuivre.

4 **Tester Mapnik**
Lancez python3 -c 'import mapnik'. La commande doit se terminer sans erreur.

2. Préparer PostgreSQL et PostGIS



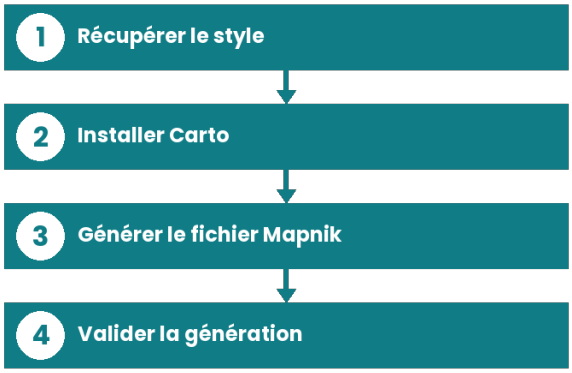
1 **Créer le rôle de rendu et la base**
Créez le rôle local _renderd, puis la base gis encodée en UTF8 et détenue par _renderd.

2 **Activer les extensions**
Dans la base gis, activez les extensions postgis et hstore.

3 **Harmoniser les propriétaires**
Attribuez les tables geometry_columns et spatial_ref_sys au rôle _renderd afin de conserver des droits cohérents.

4 **Vérifier la base**
Contrôlez les extensions avec sudo -u postgres psql -d gis -c '\dx' et le rôle avec sudo -u postgres psql -d gis -c '\du'.

3. Préparer le style openstreetmap-carto



1 **Récupérer le style**
Dans ~/src, clonez le dépôt openstreetmap-carto, récupérez ses références puis placez-vous sur la version v5.9.0.

2 **Installer Carto**
Installez Carto globalement avec npm. La version utilisée pour cette pile est Carto 1.2.0.

3 **Générer le fichier Mapnik**
Depuis le répertoire du style, exécutez carto project.mml > mapnik.xml. Ce fichier XML sera utilisé par le moteur de rendu.

4 **Valider la génération**
Vérifiez que mapnik.xml n'est pas vide avec test -s ~/src/openstreetmap-carto/mapnik.xml. Contrôlez aussi la version du style avec git -C ~/src/openstreetmap-carto describe --tags --exact-match.

! **Commencer petit réduit les risques**
Importez d'abord un extrait régional .osm.pbf. La planète entière demande des ressources nettement supérieures et ne convient pas à un premier déploiement sur un VPS classique. Le temps d'import varie notamment selon la mémoire, la taille de l'extrait et les niveaux de zoom pré-rendus.

4. Importer les données et préparer la publication

- [ ] **Télécharger l'extrait régional choisi**
L'exemple de l'article utilise une source Geofabrik ; sélectionnez la zone réellement utile.
- [ ] **Importer le fichier .osm.pbf avec osm2pgsql**
Ciblez la base gis et utilisez --create uniquement si la reconstruction des tables est acceptable.
- [ ] **Vérifier que l'import s'est terminé sans erreur**
Ne passez au rendu qu'après confirmation de la fin de l'import.
- [ ] **Relier le style, Mapnik et renderd**
Le fichier mapnik.xml généré doit être disponible pour la configuration de rendu.
- [ ] **Activer la publication via renderd, mod_tile et Apache**
Apache est le service exposé aux visiteurs ; PostgreSQL reste local au serveur dans cette procédure.
- [ ] **Tester l'affichage de quelques tuiles**
Validez le cycle complet : données importées, style compilé, rendu produit et tuiles servies par le web.

Repères à retenir

- La pile repose sur PostgreSQL 16, PostGIS, osm2pgsql, Mapnik, renderd, mod_tile et Apache.
- La cohérence des versions est importante : Ubuntu 24.04, openstreetmap-carto v5.9.0 et Carto 1.2.0 sont les repères de cette procédure.
- Le fichier mapnik.xml doit être généré avant que renderd puisse produire les tuiles.
- Gardez PostgreSQL sur localhost lorsque la base et le moteur de rendu sont sur le même serveur.
- Rejouez les commandes sur une machine vierge avant toute mise en production, car les paquets et les interfaces évoluent.

Procédure basée sur la pile et les versions décrites dans l'article. Vérifiez la documentation officielle et testez chaque commande avant une mise en production.