

supmap-devops

Authentification

NB: Nous avons spécialement ouvert notre registre d'images au public pour le rendu de ce projet. Il n'est plus nécessaire de s'authentifier pour les télécharger.

Pour déployer les conteneurs constituant le backend, il faut être authentifié par `docker login`.

- Générer un **Personal Access Token** sur GitHub :
 - Se rendre sur <https://github.com/settings/tokens>
 - Cliquer sur `Generate new token`
 - Cocher au minimum la permission `read:packages`
 - Copier le token
- Connecter Docker à GHCR avec le token :

```
echo 'YOUR_GITHUB_TOKEN' | docker login ghcr.io -u YOUR_GITHUB_USERNAME --password-stdin
```

Lancement du projet (complet)

Pour lancer le projet, il vous faudra écrire un fichier `.env` à la racine de `supmap-devops` et contenant les variables d'environnement nécessaire au bon fonctionnement du projet. Un exemple complet et fonctionnel pour un environnement local est disponible dans [.env.example](#).

Sélection de la région pour les services de routing et géocoding

Prenez soin de sélectionner la région qui vous convient le mieux pour vos tests ! Pour ce faire, définissez les variables d'environnements `PBF_NAME` et `PBF_URL` en choisissant le fichier `.pbk` correspondant avec [ce lien](#). Attention à ne pas prendre une région trop grande ! Par exemple la france entière met plusieurs heures à être indexé par le projet. Pendant ce temps, le projet n'est pas utilisable. Par exemple, pour la Basse-Normandie les variables d'environnement seront :

- `PBF_NAME=calvados.osm.pbf`
- `PBF_URL=https://download.openstreetmap.fr/extracts/europe/france/basse_normandie/${PBF_NAME}`

Lancement avec Docker Compose

Placez-vous à la racine du projet `supmap-devops` et exécutez la commande suivante :

```
docker compose up -d
```

NB: Le déploiement est très long, car Valhalla et Nominatim ont besoin d'indexer le fichier téléchargé par le conteneur `supmap-pbf-downloader`. Un health check est configuré sur ces deux services. Lorsque leur état est `Healthy`, alors le projet est totalement opérationnel.

Variables d'environnement

Variable	Type	Description
NOMINATIM_HOST	string	Nom d'hôte du service Nominatim (géocodage)
NOMINATIM_PORT	number	Port du service Nominatim
VALHALLA_HOST	string	Nom d'hôte du service Valhalla (routage)
VALHALLA_PORT	number	Port du service Valhalla
GIS_BASE_DIR	string	Chemin du répertoire contenant les fichiers GIS
PBF_NAME	string	Nom du fichier OpenStreetMap au format PBF
PBF_URL	string	URL de téléchargement du fichier PBF
REDIS_HOST	string	Nom d'hôte du service Redis
REDIS_PORT	number	Port du service Redis
REDIS_INCIDENTS_CHANNEL	string	Nom du canal Redis pour les incidents
JWT_SECRET	string	Clé secrète pour la génération/vérification des JWT
POSTGRES_USER	string	Nom d'utilisateur PostgreSQL
POSTGRES_PASSWORD	string	Mot de passe PostgreSQL
POSTGRES_PORT	number	Port PostgreSQL
POSTGRES_DB	string	Nom de la base de données principale
POSTGRES_USERS_DB	string	Nom de la base de données des utilisateurs
POSTGRES_INCIDENTS_DB	string	Nom de la base de données des incidents
USERS_DB_URL	string	URL de connexion à la base de données utilisateurs
INCIDENT_DB_URL	string	URL de connexion à la base de données incidents
CONTAINER_INTERNAL_PORT	number	Port interne des conteneurs
SUPMAP_GATEWAY_PORT	number	Port de la passerelle API
SUPMAP_USERS_HOST	string	Nom d'hôte du service utilisateurs
SUPMAP_USERS_PORT	number	Port du service utilisateurs
SUPMAP_INCIDENTS_HOST	string	Nom d'hôte du service incidents
SUPMAP_INCIDENTS_PORT	number	Port du service incidents
SUPMAP_GIS_HOST	string	Nom d'hôte du service GIS
SUPMAP_GIS_PORT	number	Port du service GIS
SUPMAP_NAVIGATION_HOST	string	Nom d'hôte du service navigation
SUPMAP_NAVIGATION_PORT	number	Port du service navigation

Services externes

Valhalla

Valhalla est un moteur de routage open source qui nous permet de :

- Calculer des itinéraires entre deux points
- Obtenir les instructions de navigation détaillées
- Optimiser les trajets en fonction de différents modes de transport (voiture, vélo, piéton)

Dans notre projet, il est principalement utilisé pour la fonctionnalité de navigation et le calcul d'itinéraires alternatifs en cas d'incidents.

Nominatim

Nominatim est un service de géocodage qui nous permet de :

- Convertir des adresses textuelles en coordonnées géographiques (géocodage)
- Convertir des coordonnées géographiques en adresses (géocodage inverse)
- Rechercher des lieux par nom ou type

Dans notre application, il est utilisé pour la recherche d'adresses et la conversion des coordonnées en adresses lisibles.

Redis / RedisInsight

Redis est une base de données en mémoire que nous utilisons pour :

- La gestion des événements en temps réel via le système pub/sub
- Le partage d'informations sur les incidents en cours
- La mise en cache de certaines données

RedisInsight est une interface graphique web qui nous permet de monitorer et gérer notre instance Redis.

Postgres

PostgreSQL est notre système de gestion de base de données (SGBD) relationnel qui stocke :

- Les données des utilisateurs (authentification, profils)
- Les informations sur les incidents (localisation, type, statut)
- L'historique des incidents

Notre architecture utilise plusieurs bases de données PostgreSQL distinctes pour séparer les différentes préoccupations (utilisateurs et incidents).