📝 TP Docker Compose – Infrastructure Complète avec Monitoring

👨‍💻 Réalisé par : [Ton prénom, nom]

📅 Date : [À compléter]

🧪 Objectif :

Mettre en place une infrastructure complète avec les services suivants :

Une application web Flask (TODO list)

Une base de données PostgreSQL

Un reverse proxy NGINX

Un monitoring avec Prometheus et Grafana

Un exporter PostgreSQL (postgres\_exporter)

# Arborescence du projet

TP-mongo/

├── docker-compose.yml

├── .env

├── flask-app/

│ ├── app.py

│ ├── requirements.txt

├── nginx/

│ └── default.conf

├── scripts/

│ └── init-db.sql

├── prometheus/

│ └── prometheus.yml

├── grafana/

│ └── [dashboard.json] (dashboard importé)

# Services Docker

## 1. Flask App (/flask-app/app.py)

**Application web en Python utilisant Flask**

**Se connecte à PostgreSQL via psycopg2**

**Routes disponibles :**

**GET /api/tasks → liste les tâches**

**POST /api/tasks → ajoute une tâche**

**GET /health → vérifie l’état de l’application**

## 2. Base de données PostgreSQL

**Image officielle postgres:15**

**Initialisation via scripts/init-db.sql**

**Contient une table tasks et un trigger pour gérer les mises à jour automatiques**

## 3. NGINX (Reverse Proxy)

**Redirige les requêtes HTTP vers le conteneur Flask**

**Fichier de conf : nginx/default.conf**

**Accès à l'app via http://localhost/api/tasks**

## 4. Prometheus

**Scrape les métriques toutes les 15 secondes**

**Voir Fichier de config : prometheus/prometheus.yml**

## 5. Postgres Exporter

Fournit les métriques PostgreSQL à Prometheus

Config via variable DATA\_SOURCE\_NAME

## 6. Grafana

* Dashboard de visualisation
* Accessible sur http://localhost:3000 (admin / admin)
* Ajout manuel d’une **data source** Prometheus
* Import d’un **dashboard JSON** depuis grafana.com

Phase test  :  
j’ai effectué des tests dont :

curl http://localhost/api/tasks # GET vide

curl -X POST http://localhost/api/tasks \

-H "Content-Type: application/json" \

-d '{"title":"Ma première tâche"}'

curl http://localhost/api/tasks # GET avec une tâche

Tests Grafana :

Connexion sur http://localhost:3000

Ajout de la source Prometheus (http://prometheus:9090)

Import d’un dashboard .json

Affichage de métriques PostgreSQL (requêtes, connexions, etc.)

# Résultat attendu

* L’API fonctionne via NGINX sur http://localhost/api/tasks
* Les tâches sont bien stockées dans la base tasks\_db
* Prometheus collecte les métriques
* Grafana affiche les données PostgreSQL sous forme de dashboard

# Sécurité & Accès

* L’application est accessible uniquement en local
* Pour un accès externe (Google), il faudrait :
  + Ouvrir les ports sur la box/firewall
  + Ou déployer via un VPS / cloud public