User-Friendly Step-by-Step Guide

1. Create a local copy of the repository

Download or clone the project to your computer.

```
root@vbox:/home/usuario1/Documents/Docker# mkdir wellnes-ops-guide
root@vbox:/home/usuario1/Documents/Docker# cd wellnes-ops-guide/
root@vbox:/home/usuario1/Documents/Docker/wellnes-ops-guide# git clone https://github.com/luisrodvillada/wellnes-ops.git
Cloning into 'wellnes-ops'...
remote: Enumerating objects: 7664, done.
remote: Counting objects: 100% (7664/7664), done.
remote: Compressing objects: 100% (5650/5650), done.
remote: Total 7664 (delta 1727), reused 7509 (delta 1572), pack-reused 0 (from 0)
Receiving objects: 100% (7664/7664), 17.88 MiB | 18.07 MiB/s, done.
Resolving deltas: 100% (1727/1727), done.
root@vbox:/home/usuario1/Documents/Docker/wellnes-ops-guide#
```

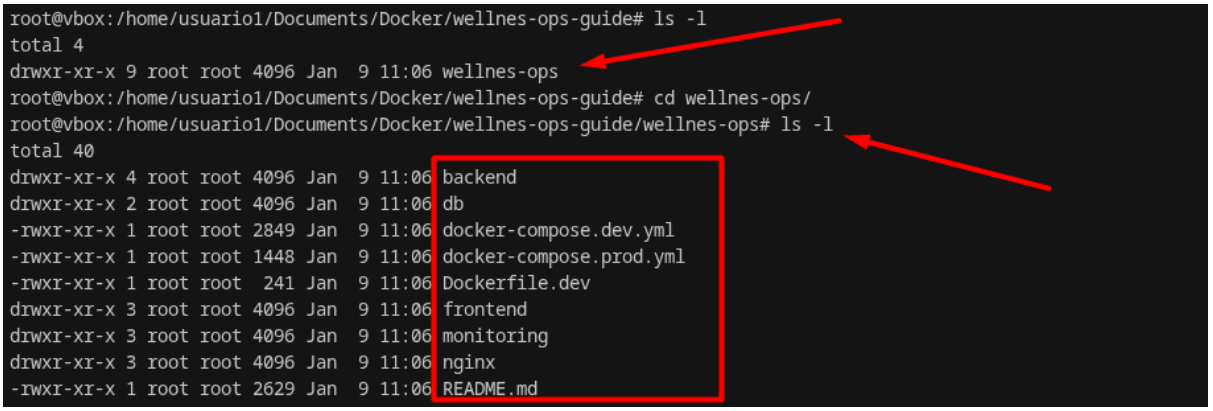
2. Prepare the environment variables

Create the required `.env` file(s) with the configuration values needed for the application to run.

```
 GNU nano 7.2                                    env/dev/.env *
POSTGRES_DB=wellness
POSTGRES_USER=postgres
POSTGRES_PASSWORD=postgres
POSTGRES_PORT=5432
POSTGRES_HOST=postgres

JWT_SECRET=super-secret-key-change-me
JWT_EXPIRES_IN=1h
```

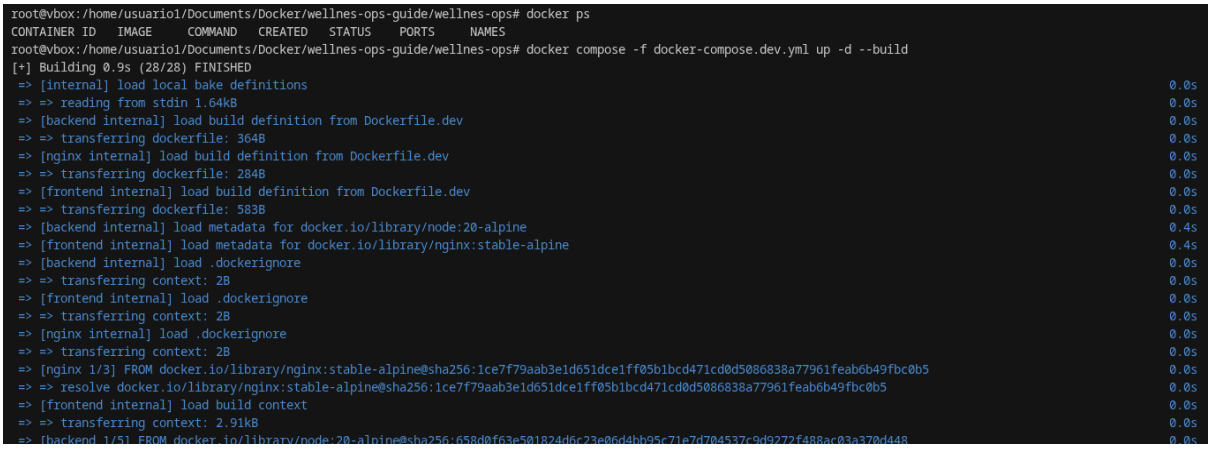
3. Review the project structure

Check the folders and files included in the project to understand how the stack is organized



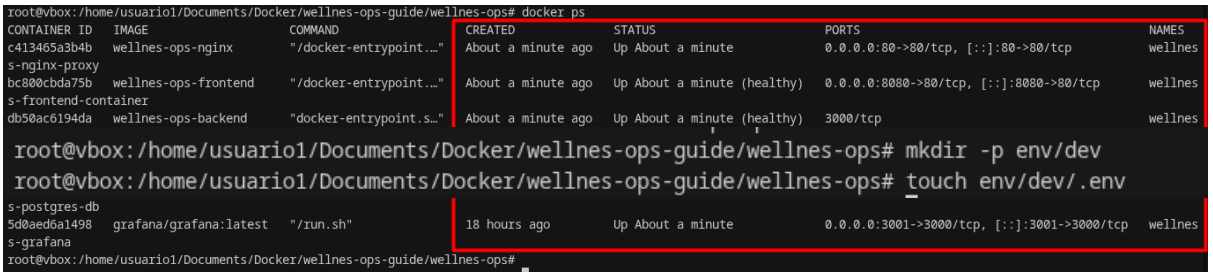
4. Start the container stack

Run the Docker Compose command to build and start all services.



5. Verify that all services are healthy

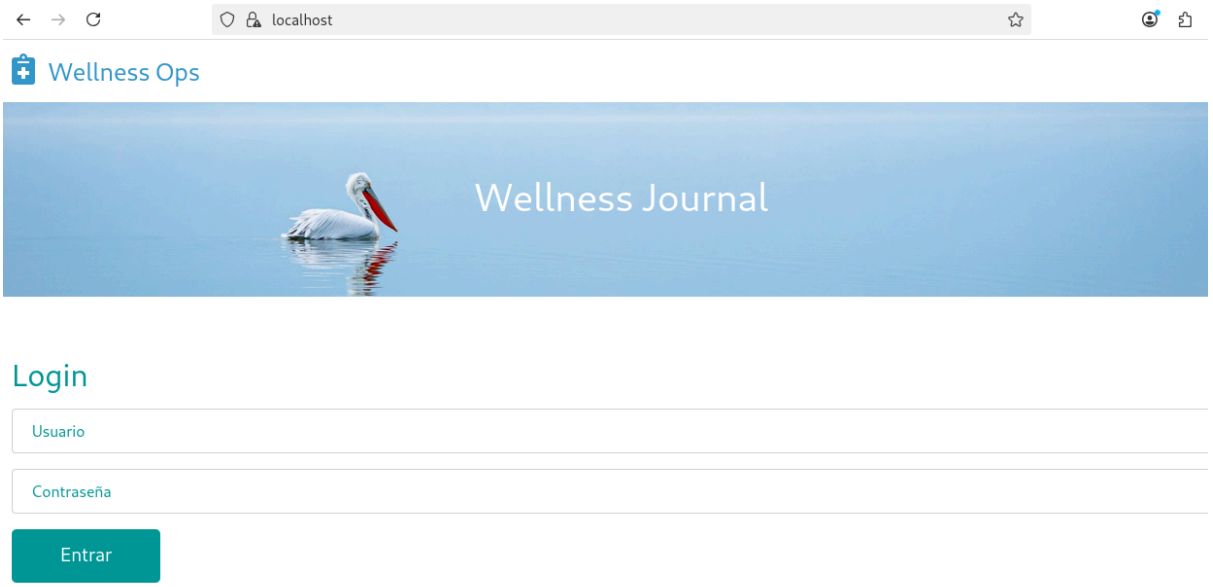
Check the container status to ensure every service is running and marked as “healthy”.



6. Open the application in your browser

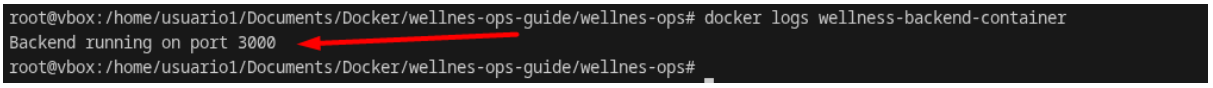
Go to:
https://localhost/

This loads the main frontend interface.



7. Check the Backend container

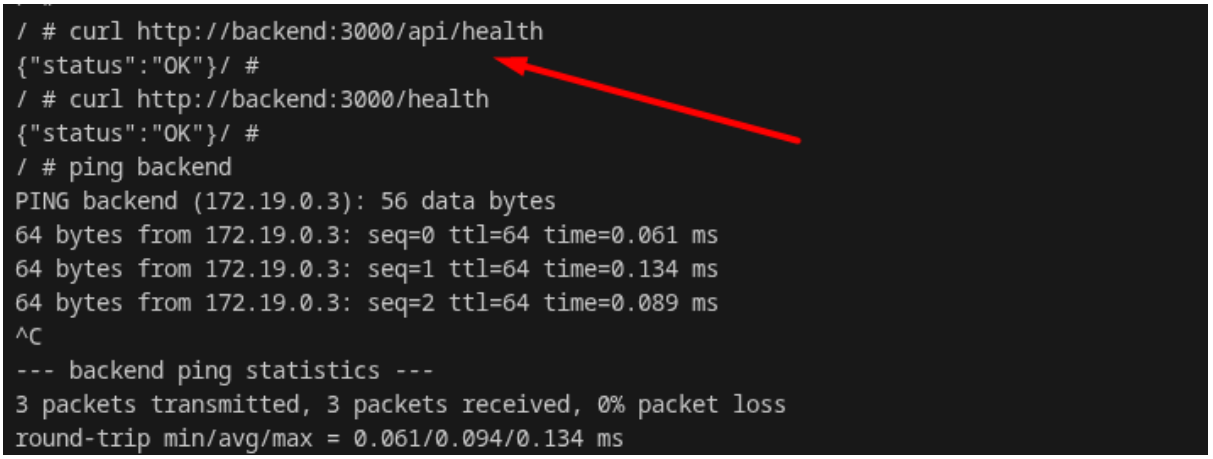
Confirm that the backend service is running correctly inside Docker.



8. Test the Backend endpoint through Nginx

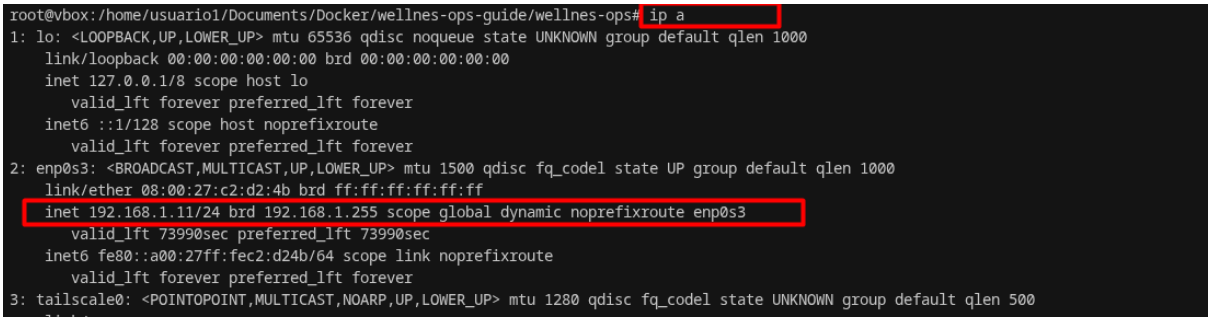
Important: The backend is **not exposed directly**.

It is only accessible **through Nginx**.



9. Check your host machine's IP address

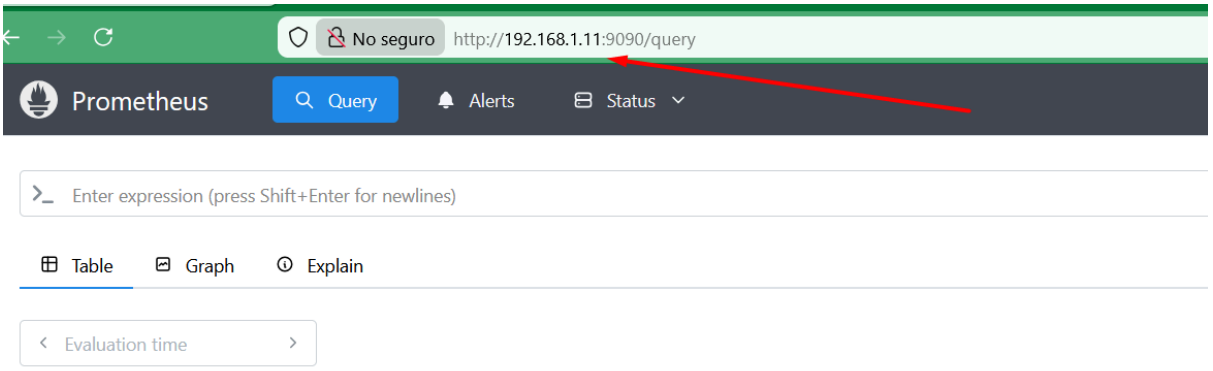
Identify your local IP (for example: `192.168.1.11:3001`) to access services from another device if needed.

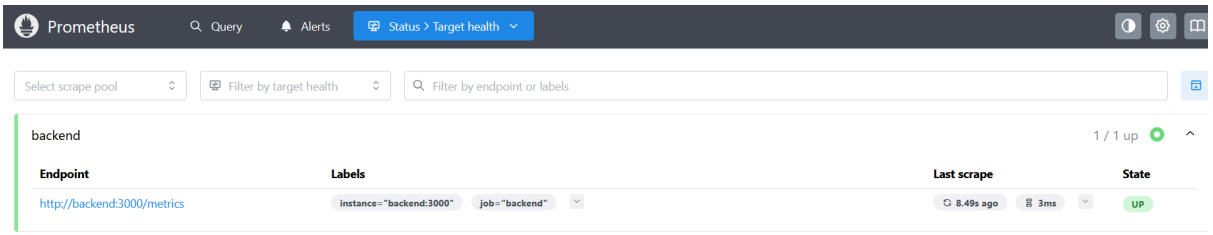


10. Access Prometheus (port 9090)

Open:
http://localhost:9090

Prometheus collects metrics from the backend.

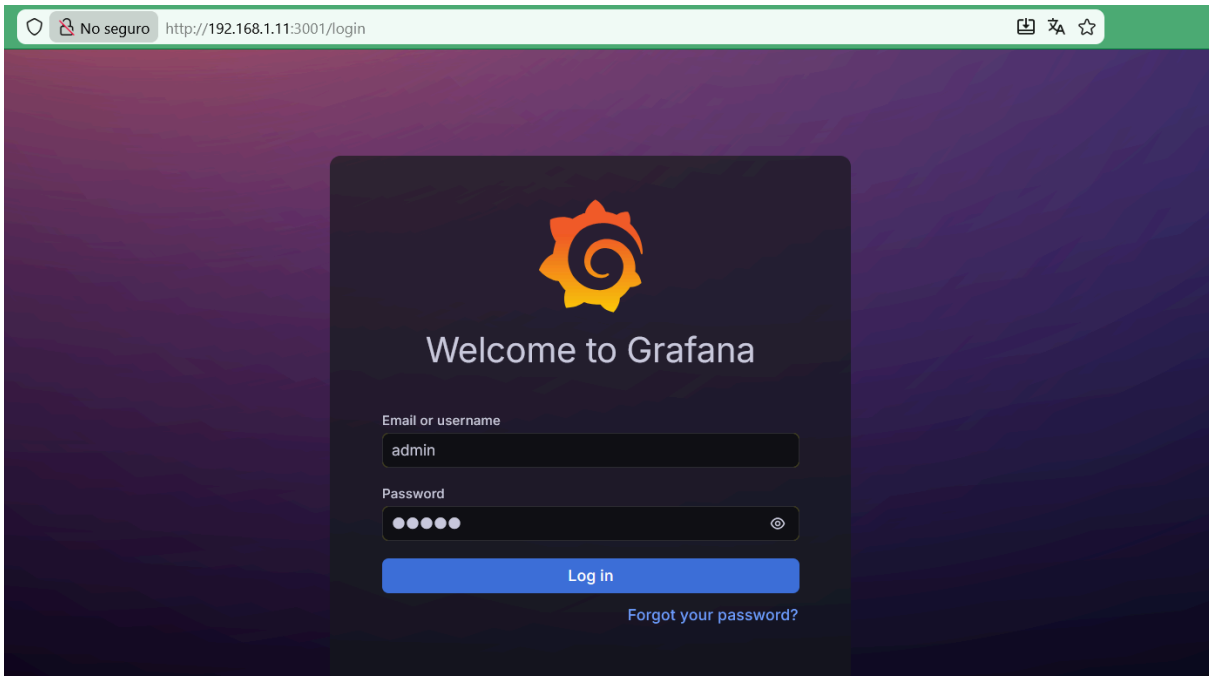




12. Access Grafana

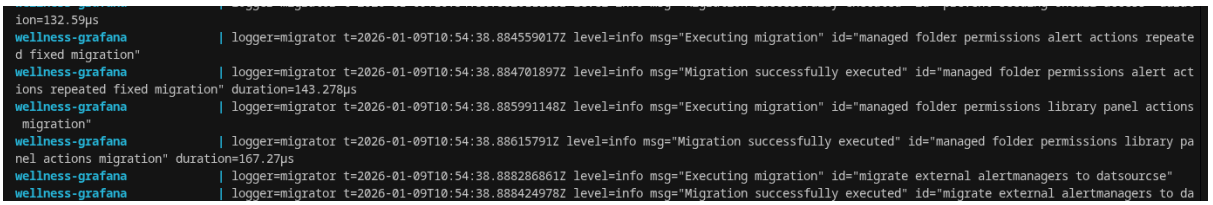
Open:
http://localhost:3001

Log in and view dashboards or create new ones.



13. View real-time logs

Use Docker logs to monitor all services in real time.



14. Access the database and inspect tables

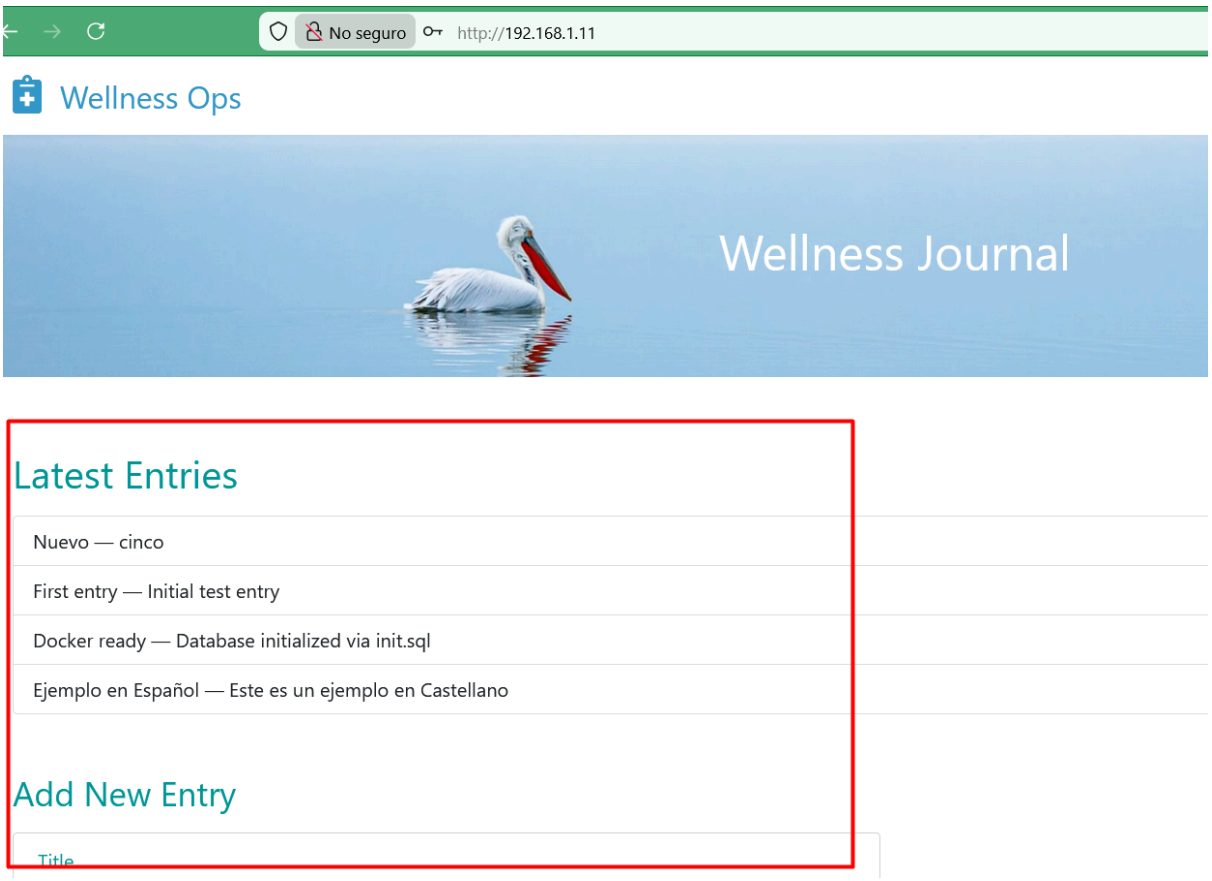
Connect to PostgreSQL and check that tables and records exist.

```
wellness=#
wellness=# SELECT * FROM entries;
 id |       title        |            description            |        created_at
----+--------------------+-----------------------------------+--------------------------
  1 | First entry        | Initial test entry                | 2026-01-08 16:13:28.66163
  2 | Docker ready       | Database initialized via init.sql | 2026-01-08 16:13:28.66163
  3 | Ejemplo en Español | Este es un ejemplo en Castellano  | 2026-01-08 16:13:28.66163
  4 | Nuevo              | cinco                             | 2026-01-08 16:21:58.6686
(4 rows)

wellness=#
```

15. Verify data persistence in the application

Create or modify data in the app and confirm that it remains stored in the database.

```
wellness=# SELECT * FROM entries;
 id |       title        |            description            |        created_at
----+--------------------+-----------------------------------+--------------------------
  1 | First entry        | Initial test entry                | 2026-01-08 16:13:28.66163
  2 | Docker ready       | Database initialized via init.sql | 2026-01-08 16:13:28.66163
  3 | Ejemplo en Español | Este es un ejemplo en Castellano  | 2026-01-08 16:13:28.66163
  4 | Nuevo              | cinco                             | 2026-01-08 16:21:58.6686
(4 rows)

wellness=#
```



16. Confirm that the CI/CD pipeline runs automatically

Push changes to the repository and verify that the pipeline executes successfully.