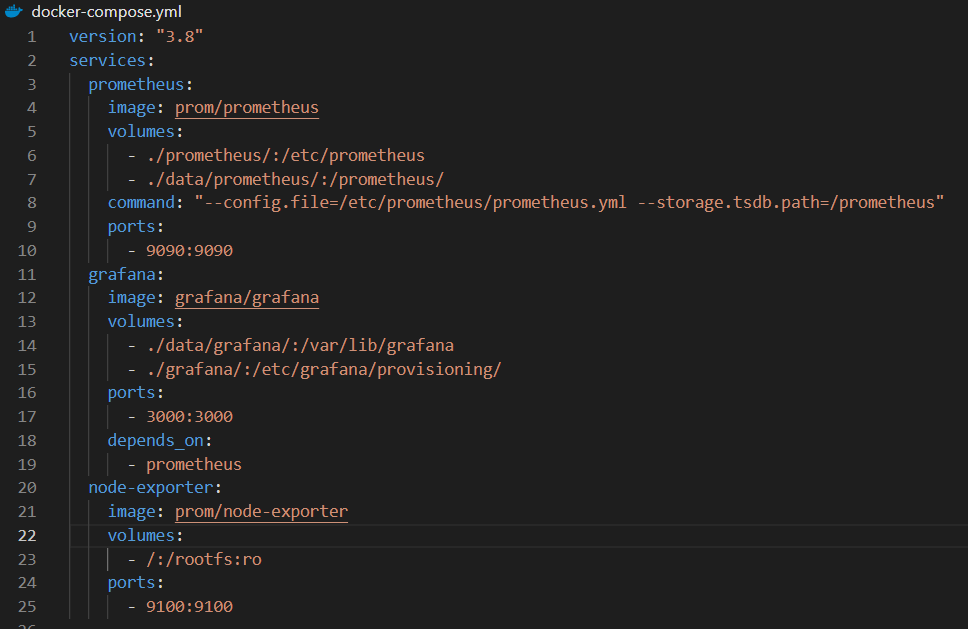
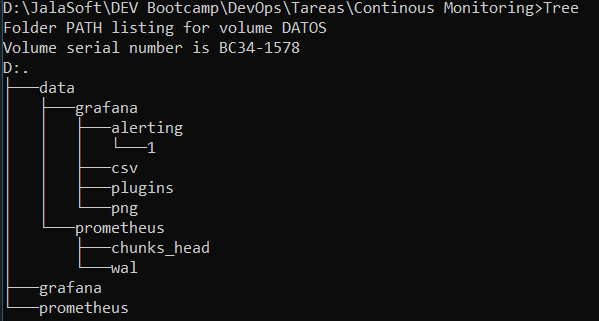
**Docker-compose, Prometheus, Grafana and Node-exporter**

In order to be able to deploy many services we can use docker compose, in this case our first task is to create docker-compose.yml file. The content is shown in the next image:

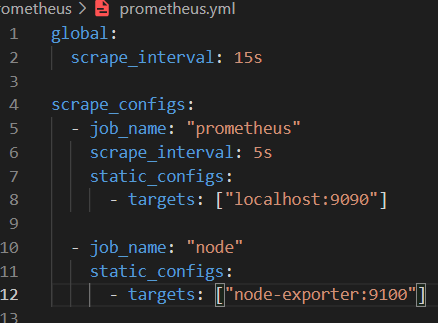


Here we define three services, one for Prometheus where we specify the image from docker hub, the volumes that are going to be created to persist data, the commands to point Prometheus config file and where the data is going to be stored, and the port binding. The next service is Grafana, we indicate the image, the volumes and port, and this service is going to be deployed after Prometheus service is up. Finally, we have node-exporter, where we specify the image, the volume and the port binding.

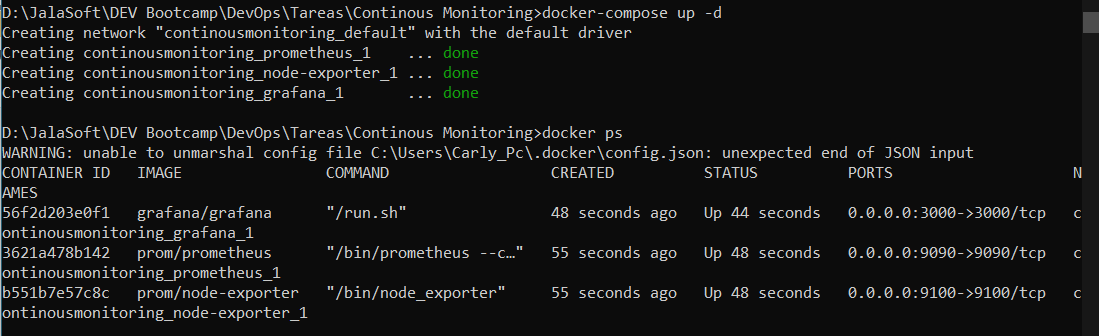
If we want the later docker-compose.yml to work it is necessary to have the next file structure, excluding the information inside /data/grafana and /data/Prometheus because it is going to be generated when we run the services:



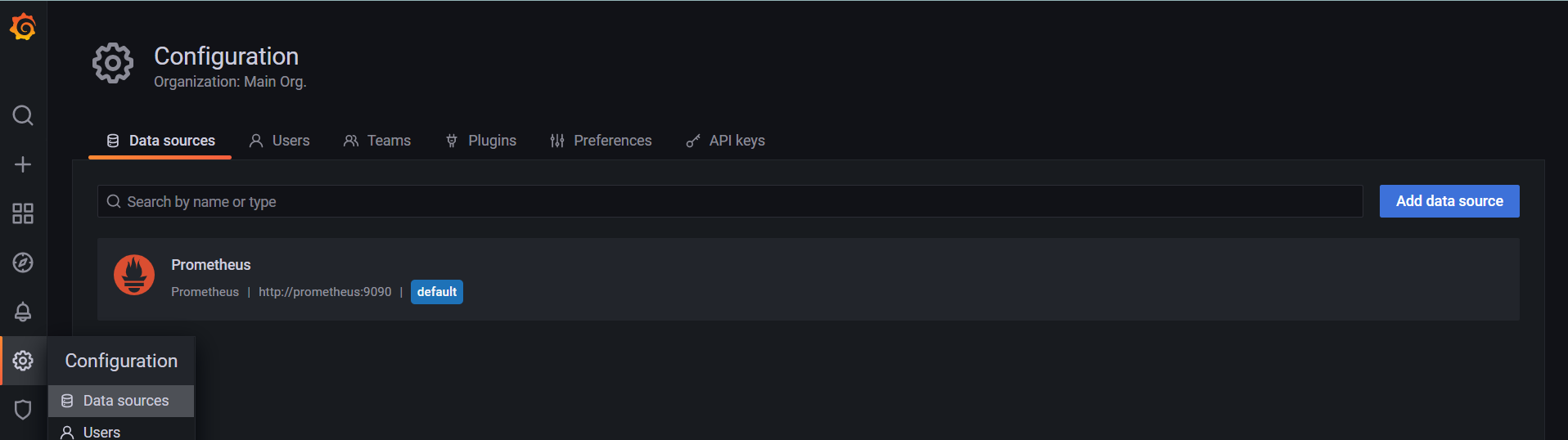
It is also necessary to have the configuration file for Prometheus prometheus.yml inside /prometheus folder with the next content:

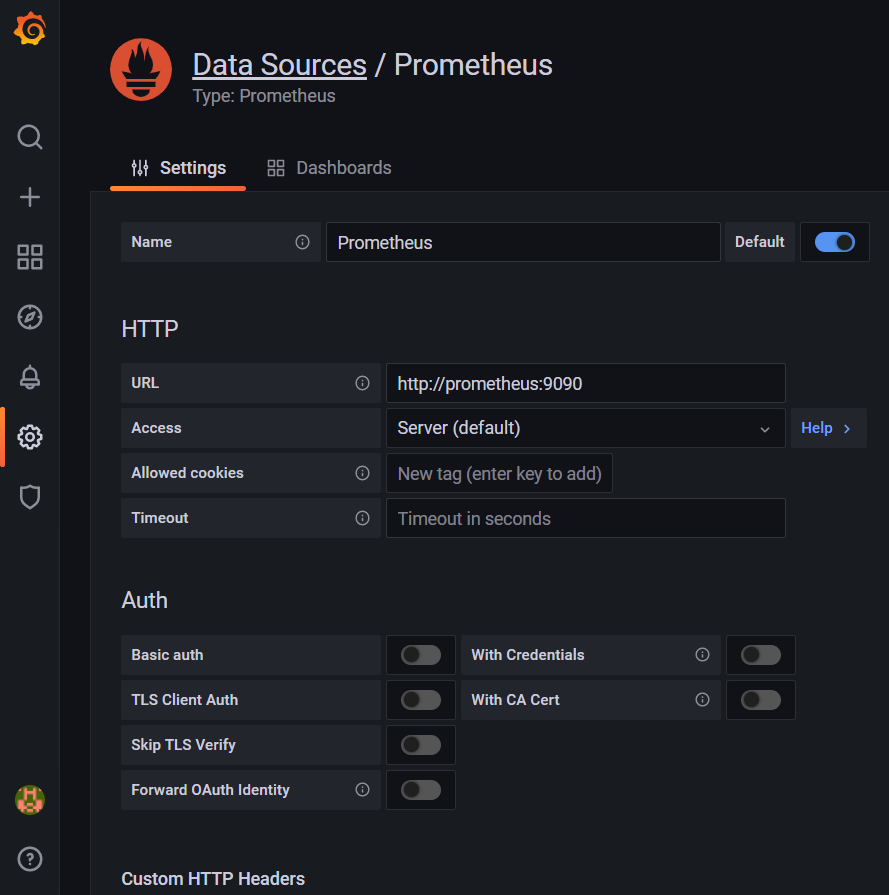


To start services, we run the command docker-compose up -d inside the folder where our docker-compose.yml is located:

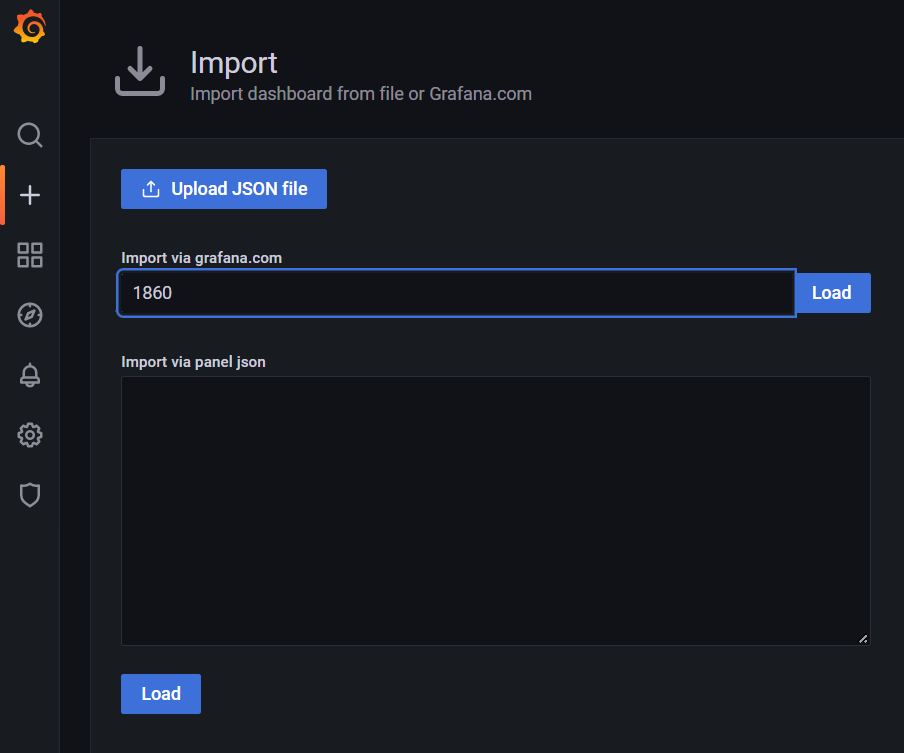
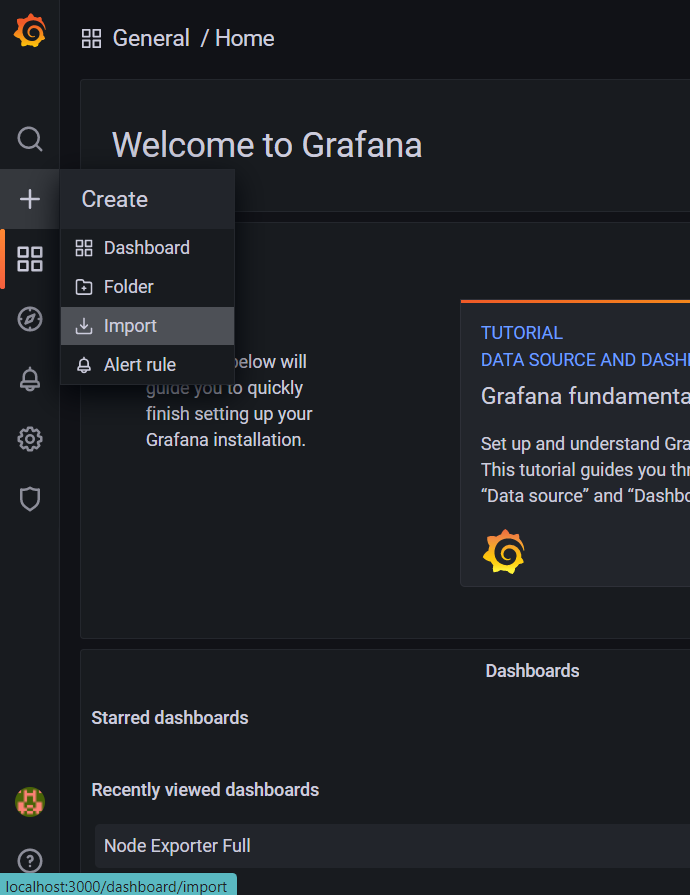


Once we have our services running we can access Prometheus in localhost:9090, node-exporter in localhost:9100 and Grafana in localhost:3000. The user and password for Grafana is admin/admin. We need to configure the Data Sources in Grafana, in this case we added Prometheus and we specify the URL.





Finally, we can import a dashboard, choosing the next options and fill the ID of node-exporter 1860:



Then we will be able to see different graphs that represent the information being monitored by Prometheus.

