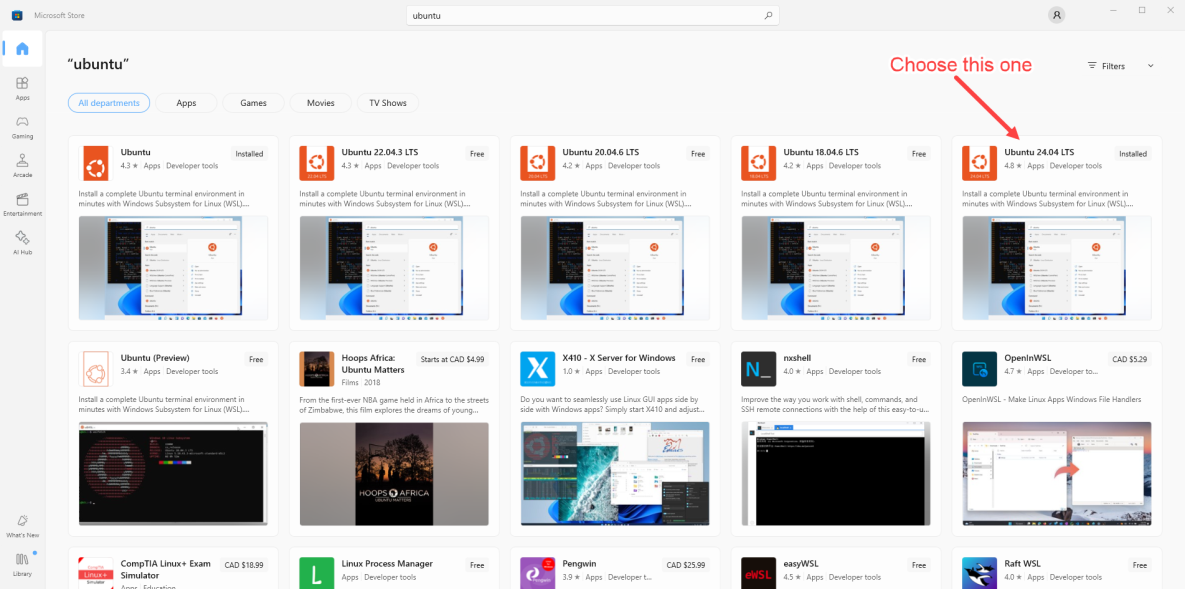
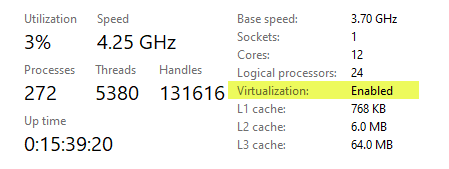
1. Developing on Windows OS, we first need to install a full version of the Linux OS, we need this to create and build Docker images and containers. Windows store comes with a flavor of Ubuntu LTS linux. We can run what is called Windows Subsystem Linux (WSL) OS on windows and keep our development on Windows
   1. Go to Windows Store, search for “ubuntu” and choose WSL Ubuntu 24.04 LTS, this is the latest long term support version at the time of this writing.

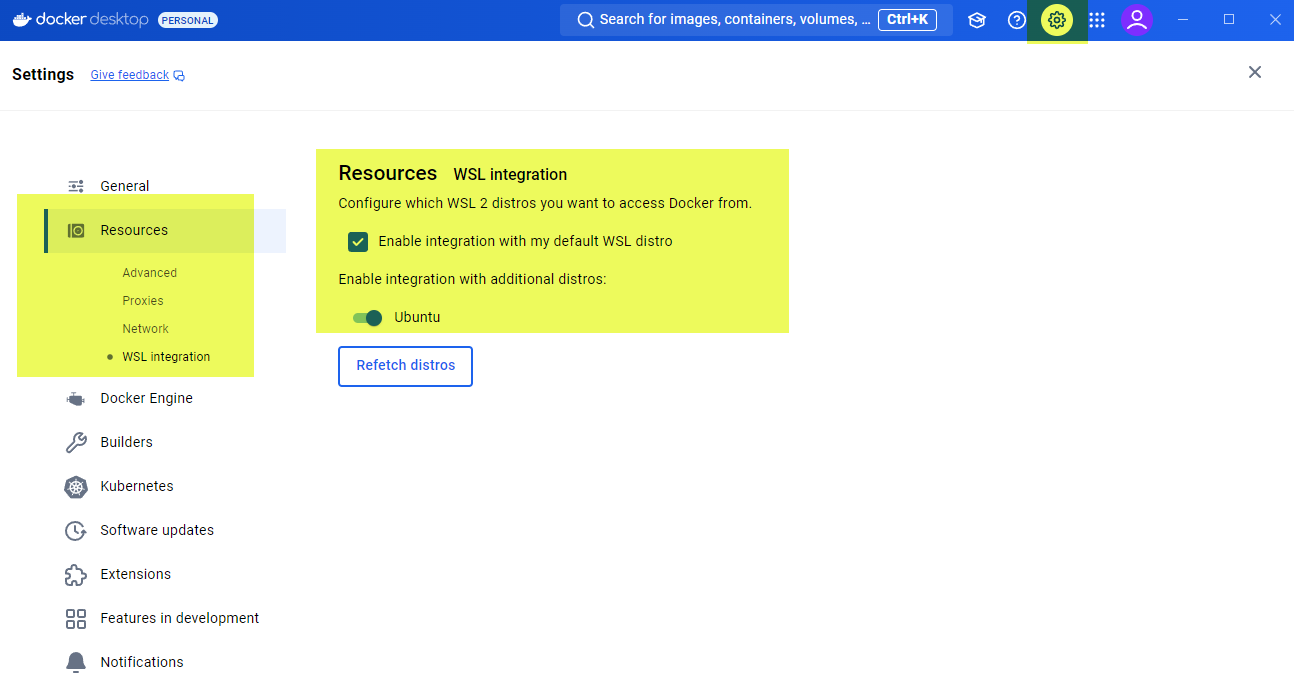


* 1. Install the Ubuntu WSL from the Windows Store

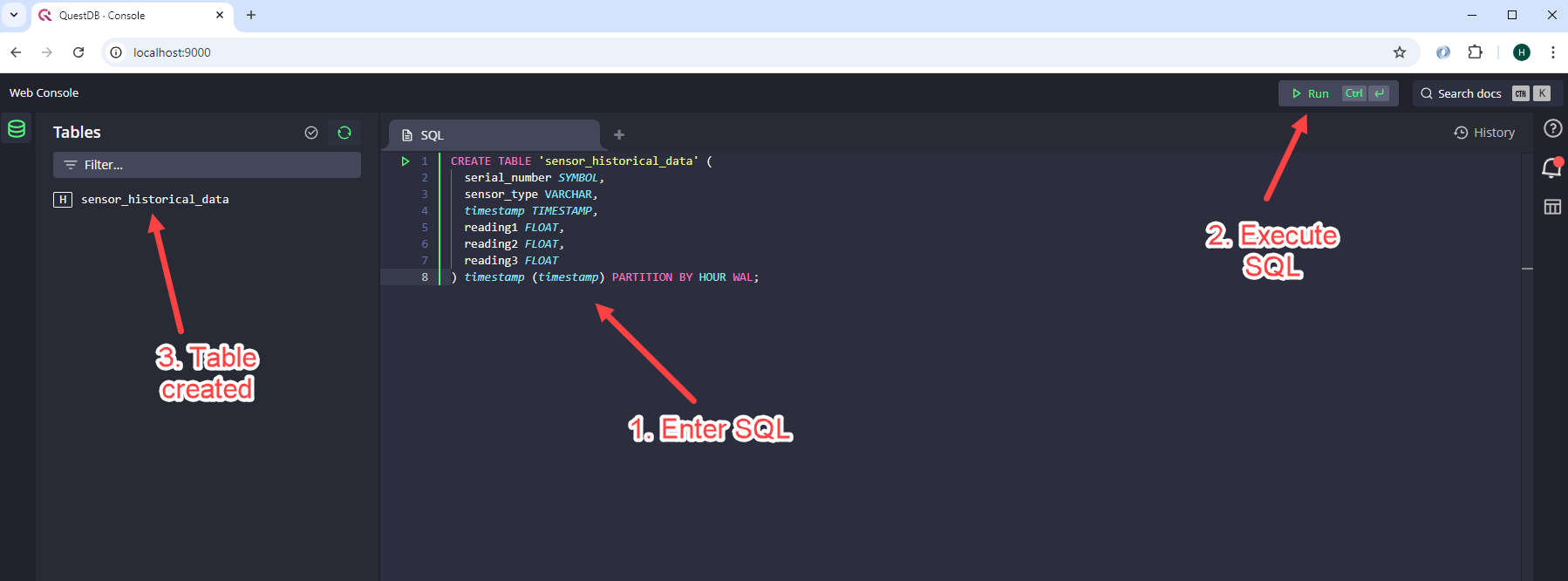
1. Download Docker Desktop for Windows OS version here -> [Docker Desktop: The #1 Containerization Tool for Developers | Docker](https://www.docker.com/products/docker-desktop/)
   1. Make sure you set your Windows machine to enable Virtualization, to verify, open Task Manager, under Performance tab, you should see the following:



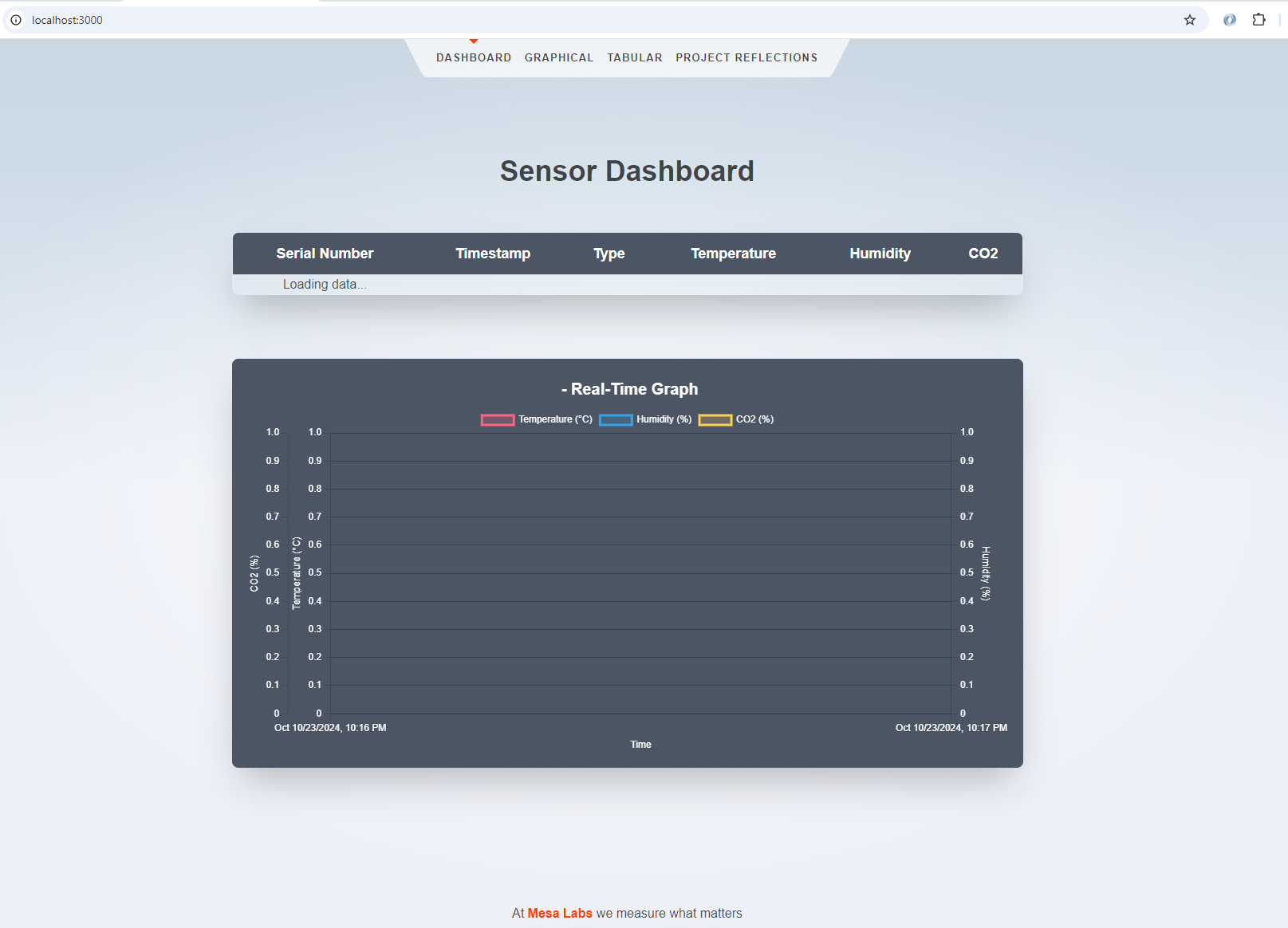
* 1. Install Docker Desktop, when ask if you want to use Hyper V or WSL, choose WSL because we are using Ubuntu WSL, this will give us a development environment that is similar to our production environment.
  2. Run Docker Desktop
  3. Go to Settings->Resources->WSL Integration, and enable integration for WSL distro, in our case it’s Ubuntu as shown below:



1. Keep Docker Desktop open and make sure WSL is running by restarting your Windows
2. Install development IDE of your choice, VS Code is good, please go here to get the latest version of VS Code IDE -> https://code.visualstudio.com/
3. Clone this project from github from the main branch
4. Open VS Code, under “**File**” -> “**Open Folder**”, choose the folder of clone project
5. Under “**Terminal**” -> “**New Terminal**”, enter the docker command: **docker compose run --entrypoint="" frontend npm install**
6. Step 7 will be a few minutes depending on your internet speed as it is downloading all the necessary node modules.
7. Once done, you should see a folder called “node\_modules” under the “frontend” folder which is the Sveltekit front end application.
8. In the **Terminal,** run the docker command: **docker compose up –build**
9. Open the browser and enter the url “**localhost:9000**”, this will bring up QuestDB web interface, enter the following SQL to create the “**sensor\_historical\_data**” table, click on the “**Run**” button at the top left to create the table as shown below:
10. CREATE TABLE 'sensor\_historical\_data' (
11. serial\_number *SYMBOL*,
12. sensor\_type VARCHAR,
13. *timestamp* *TIMESTAMP*,
14. reading1 *FLOAT*,
15. reading2 *FLOAT*,
16. reading3 *FLOAT*
17. ) *timestamp* (*timestamp*) PARTITION BY HOUR WAL;

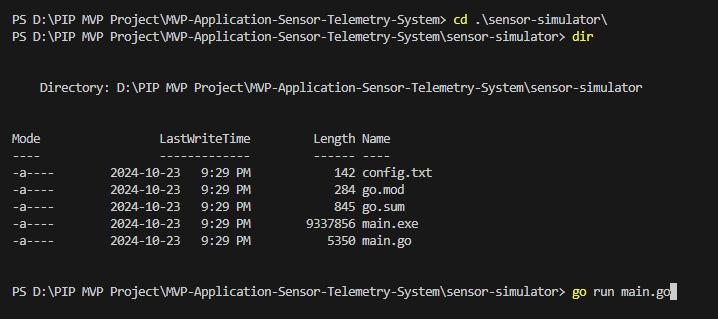
****

**12.** Open another browser tab and enter the url “localhost:3000”, this is the Svelte application being serve on port 3000, you should see the following:



13. At this point all docker services are all up and running, Kafka broker, Backend service written in Golang, QuestDB database, and front end application Sveltekit app. We need to run the sensor simulator program written in Golang to see data flowing in. See next step.

14. In VS Code IDE, open another terminal, you should already be in the folder of the project already, change directory to “sensor-simulator” as shown below. Then enter the command: **go run main.go,** the simulator will run and you can watch the sensor data coming in on the web browser at the url: **localhost:3000**

****