# Running Ollama llama3 on WSL Ubuntu using docker

It appears that Docker is not installed on your system or it's not correctly configured. Let's proceed with installing Docker [1]:

## Installing Docker on Ubuntu

### Update Package Index:

sudo apt-get update

### Install Dependencies:

sudo apt-get install \

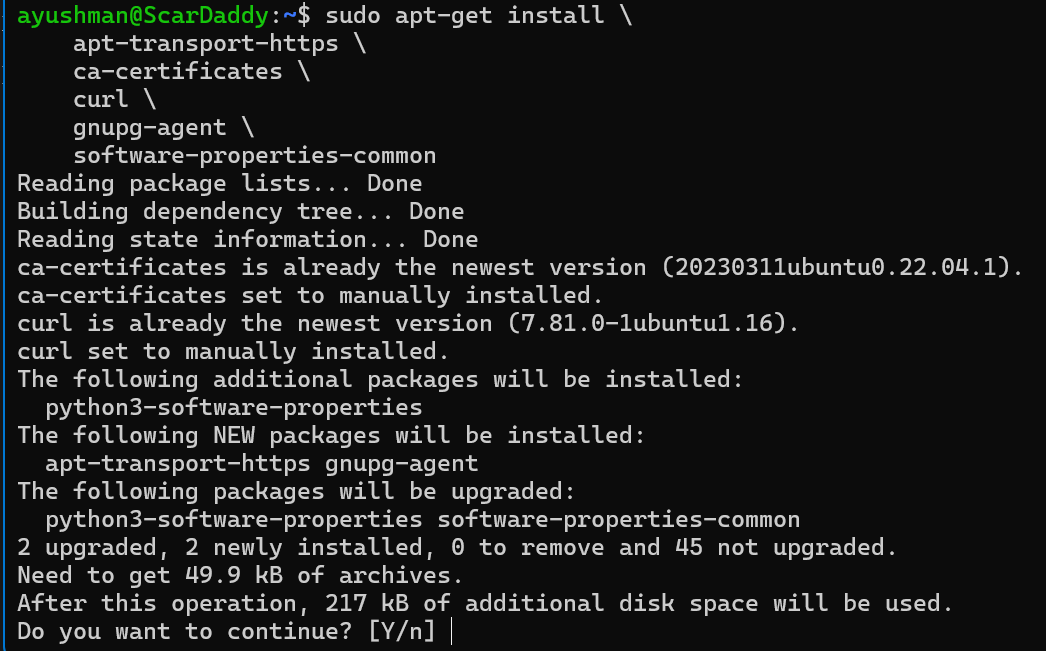
apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common



### Setting Up the Docker Repository

Since apt-key is deprecated, we'll follow the updated method to add the GPG key and the Docker repository.

#### **Create the keyring directory (if it does not exist):**

sudo mkdir -p /etc/apt/keyrings



#### **Add the Docker GPG key using the updated method:**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

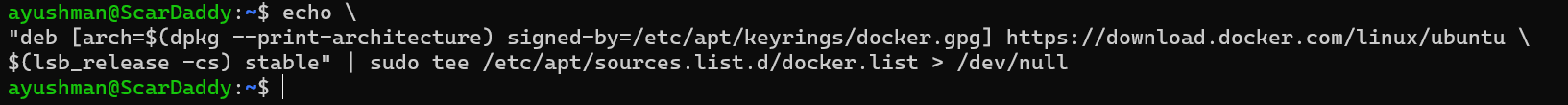


#### **Add the Docker repository to your APT sources:**

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

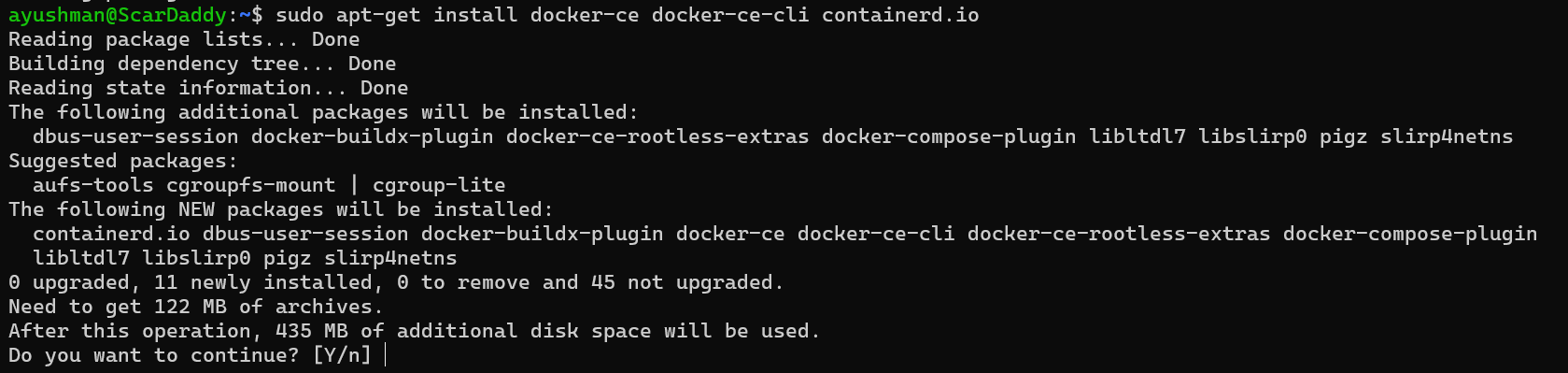


#### **Update the package index:**

sudo apt-get update

#### **Install Docker Engine:**

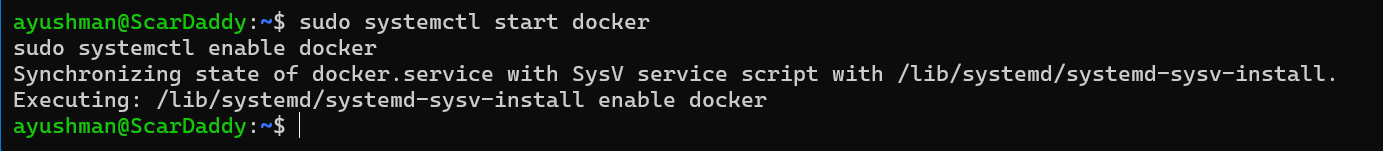
sudo apt-get install docker-ce docker-ce-cli containerd.io



#### **Start and enable Docker:**

sudo systemctl start docker

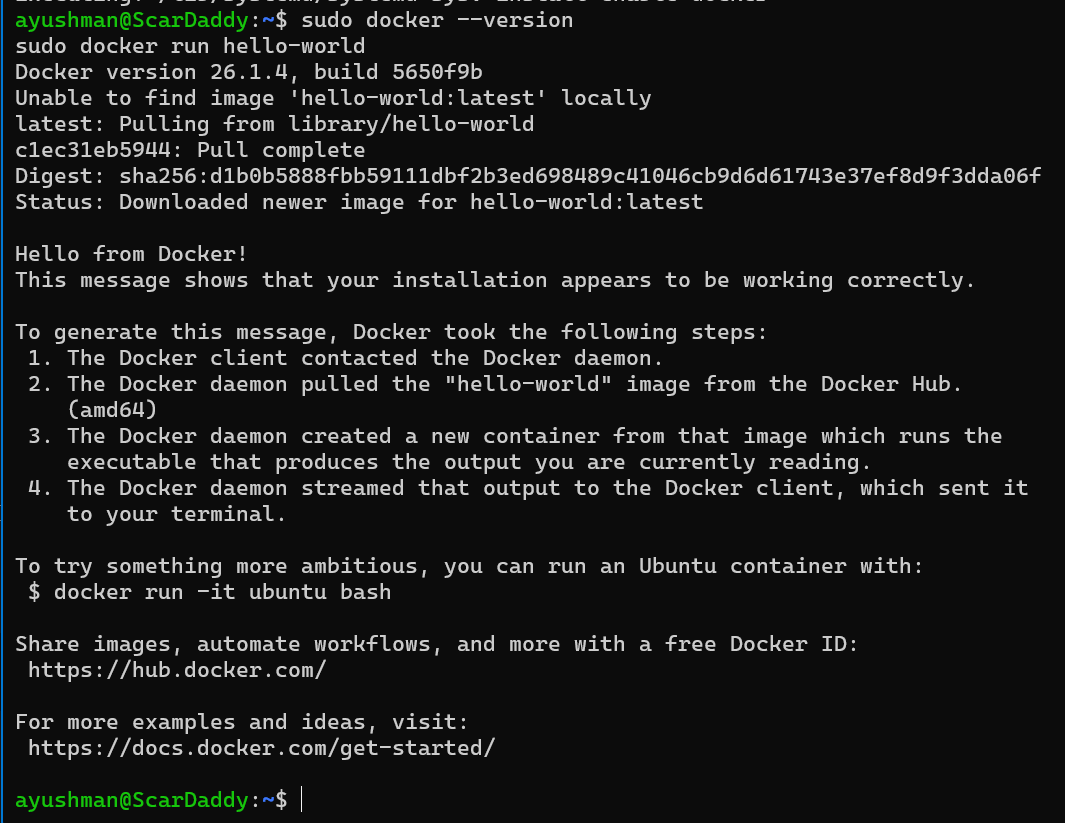
sudo systemctl enable docker



#### **Verify the Docker installation:**

sudo docker --version

sudo docker run hello-world

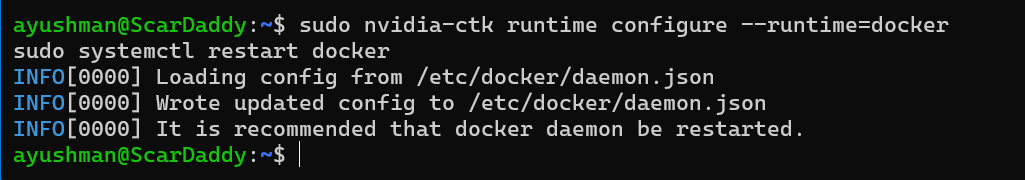


Following these steps should ensure that Docker is installed correctly on your system. If you encounter any issues, please share the specific error messages, and I'll assist you further.

## **Configure Docker to use Nvidia driver**

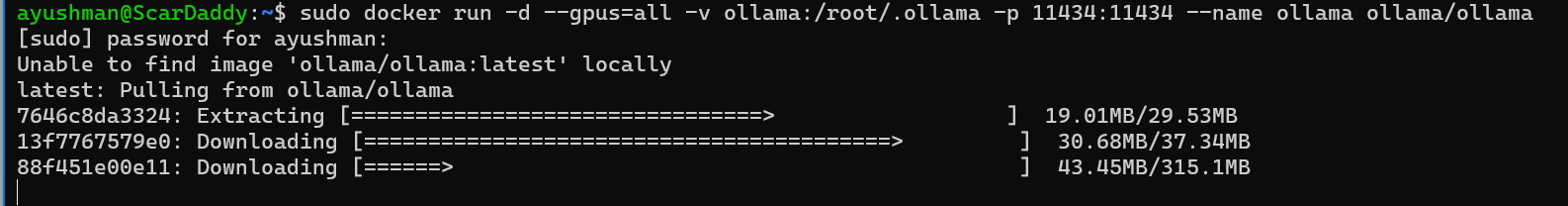
sudo nvidia-ctk runtime configure --runtime=docker

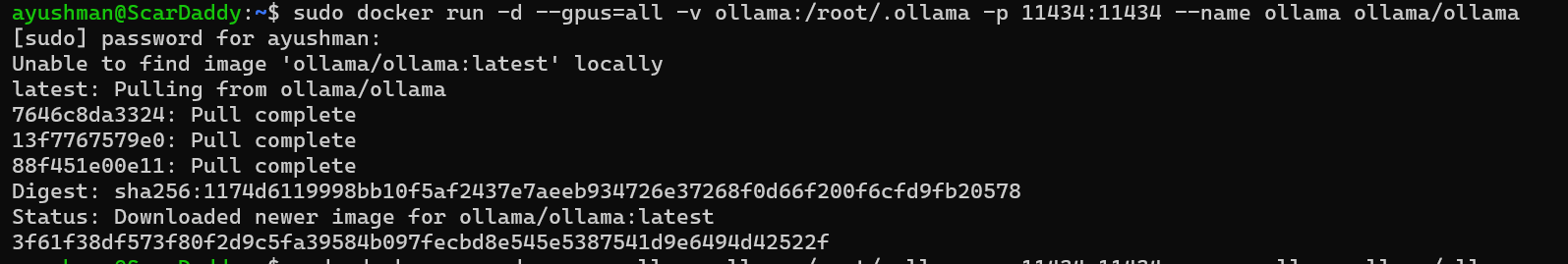
sudo systemctl restart docker

****

## **Start the container**

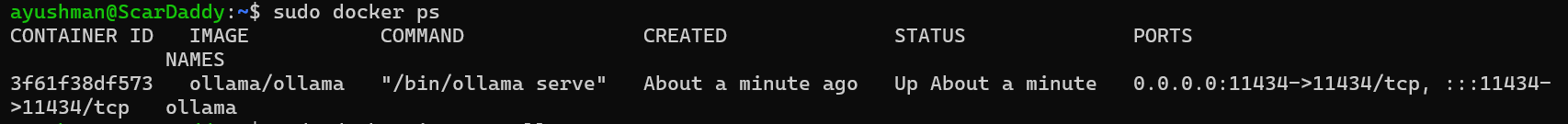
Sudo docker run -d --gpus=all -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama





It looks like the Docker container named "ollama" is already running. You can check the status of running Docker containers using the following command:

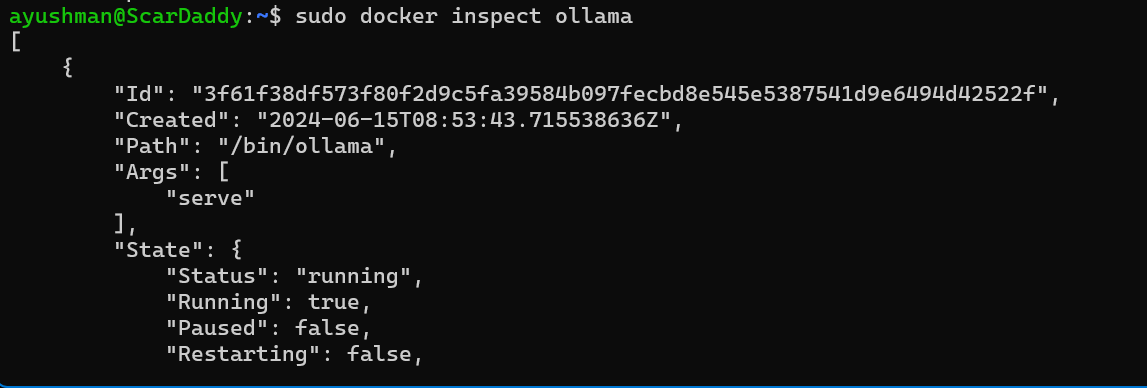
sudo docker ps



This will list all currently running containers. Look for the container with the name "ollama" in the output to confirm that it is running.

If you need more detailed information about the container, you can use:

sudo docker inspect ollama



This will give you detailed information about the container, including its current status and configuration.

If you need to stop the container, you can use:

sudo docker stop ollama

And to remove the container:

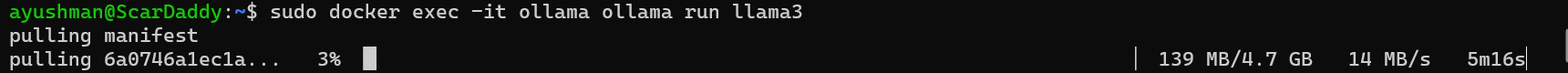
sudo docker rm ollama

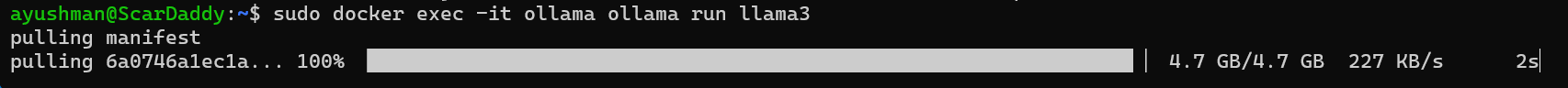
After removing the container, you can run it again without encountering the name conflict error.

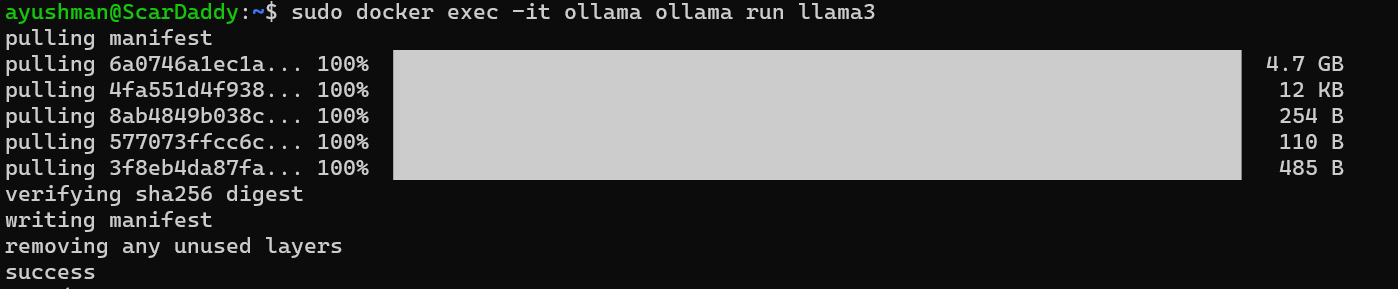
## Run model locally

Now you can run a model:

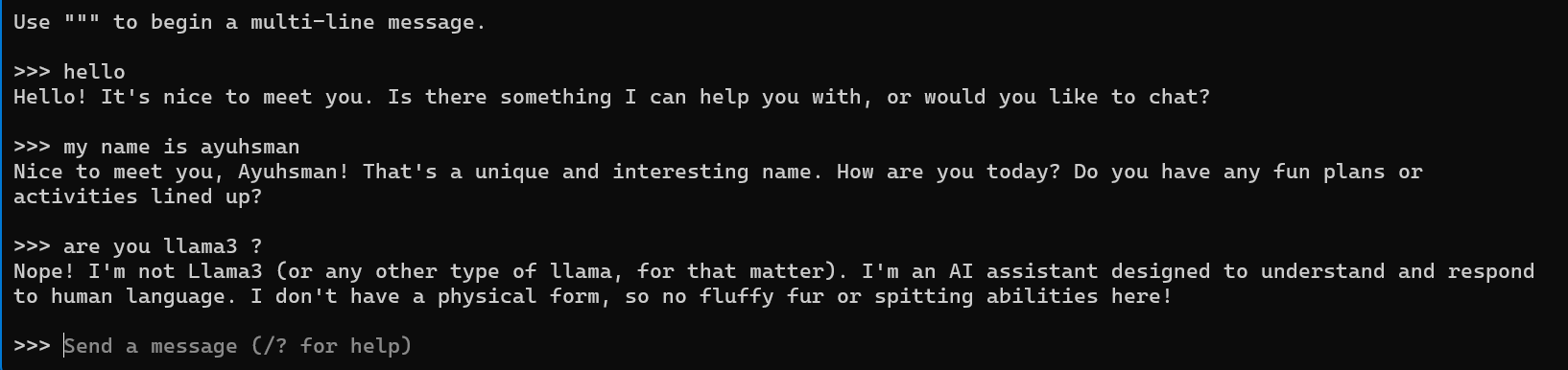
docker exec -it ollama ollama run llama3







## Start prompting



## How to run this using python?

pip install ollama

from ollama import Client

def query\_llama3(question):

client = Client(host='http://localhost:11434')

response = client.chat(model='llama3', messages=[

{

'role': 'user',

'content': question,

},

])

return response

if \_\_name\_\_ == "\_\_main\_\_":

question = "Why is the sky blue?"

response = query\_llama3(question)['message']['content']

print(response)

INFO:httpx:HTTP Request: POST http://localhost:11434/api/chat "HTTP/1.1 200 OK"

What a great question!

The sky appears blue because of a phenomenon called Rayleigh scattering, named after the British physicist Lord Rayleigh. Here's what happens:

1. \*\*Sunlight enters Earth's atmosphere\*\*: When sunlight travels through space, it contains all wavelengths (colors) of visible light.

2. \*\*Atmosphere scatters shorter wavelengths\*\*: As sunlight enters our atmosphere, it encounters tiny molecules of gases like nitrogen (N2), oxygen (O2), and others. These molecules are much smaller than the wavelength of light.

3. \*\*Blue light is scattered more\*\*: The small molecules scatter shorter wavelengths of light, such as blue and violet, more efficiently than longer wavelengths, like red and orange. This is because the smaller molecules are better at interacting with these shorter wavelengths.

4. \*\*Red light continues straight\*\*: Meanwhile, the longer wavelengths of light, like red and orange, continue to travel in a straight line, reaching our eyes from a shorter distance.

5. \*\*Our eyes perceive the scattered blue light\*\*: As we look up at the sky, we see the combined effect of all this scattering: more blue light being scattered in all directions, making the sky appear blue.

The exact shade of blue can vary depending on several factors:

\* Time of day: The sky tends to be more blue during sunrise and sunset when the sun is lower in the sky.

\* Atmospheric conditions: Pollution, dust, and water vapor can scatter light differently, changing the apparent color of the sky.

\* Altitude: At higher altitudes, the air pressure is lower, which can affect the way light scatters.

So, to summarize, the sky appears blue because the tiny molecules in our atmosphere scatter shorter wavelengths of sunlight, like blue and violet, more efficiently than longer wavelengths.

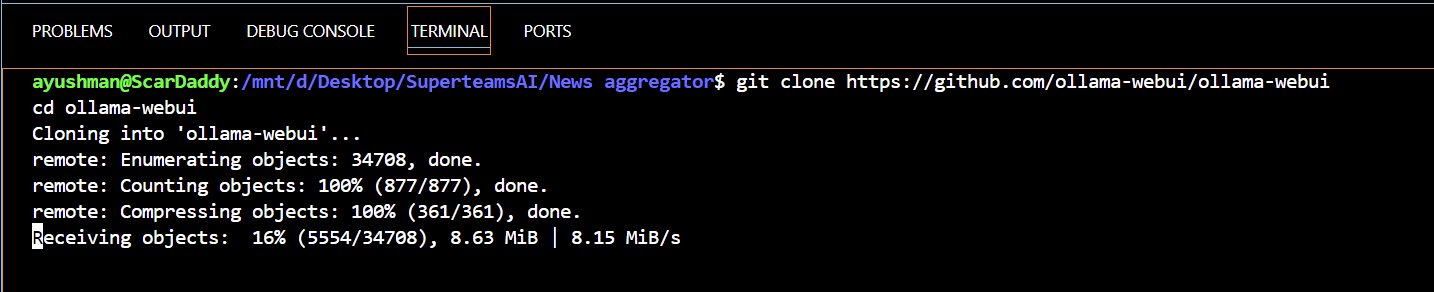
Congratulations, you’re now connected to the power of Llama 3!

## Setting up Ollama WebUI

Clone the official repository of Ollama WebUI

git clone https://github.com/ollama-webui/ollama-webui

cd ollama-webui



Open up the Compose file to see the YAML file:

version: '3.6'

services:

ollama:

volumes:

- ollama:/root/.ollama

# Uncomment below to expose Ollama API outside the container stack

# ports:

# - 11434:11434

container\_name: ollama

pull\_policy: always

tty: true

restart: unless-stopped

image: ollama/ollama:latest

ollama-webui:

build:

context: .

args:

OLLAMA\_API\_BASE\_URL: '/ollama/api'

dockerfile: Dockerfile

image: ollama-webui:latest

container\_name: ollama-webui

depends\_on:

- ollama

ports:

- 3000:8080

environment:

- "OLLAMA\_API\_BASE\_URL=http://ollama:11434/api"

extra\_hosts:

- host.docker.internal:host-gateway

restart: unless-stopped

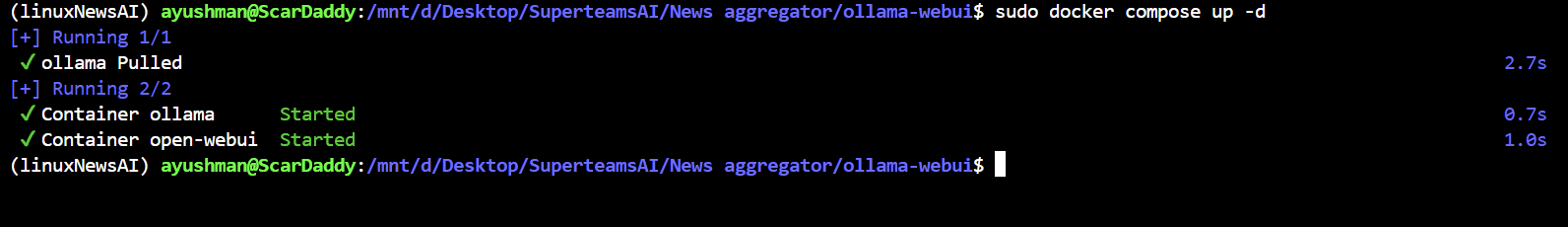
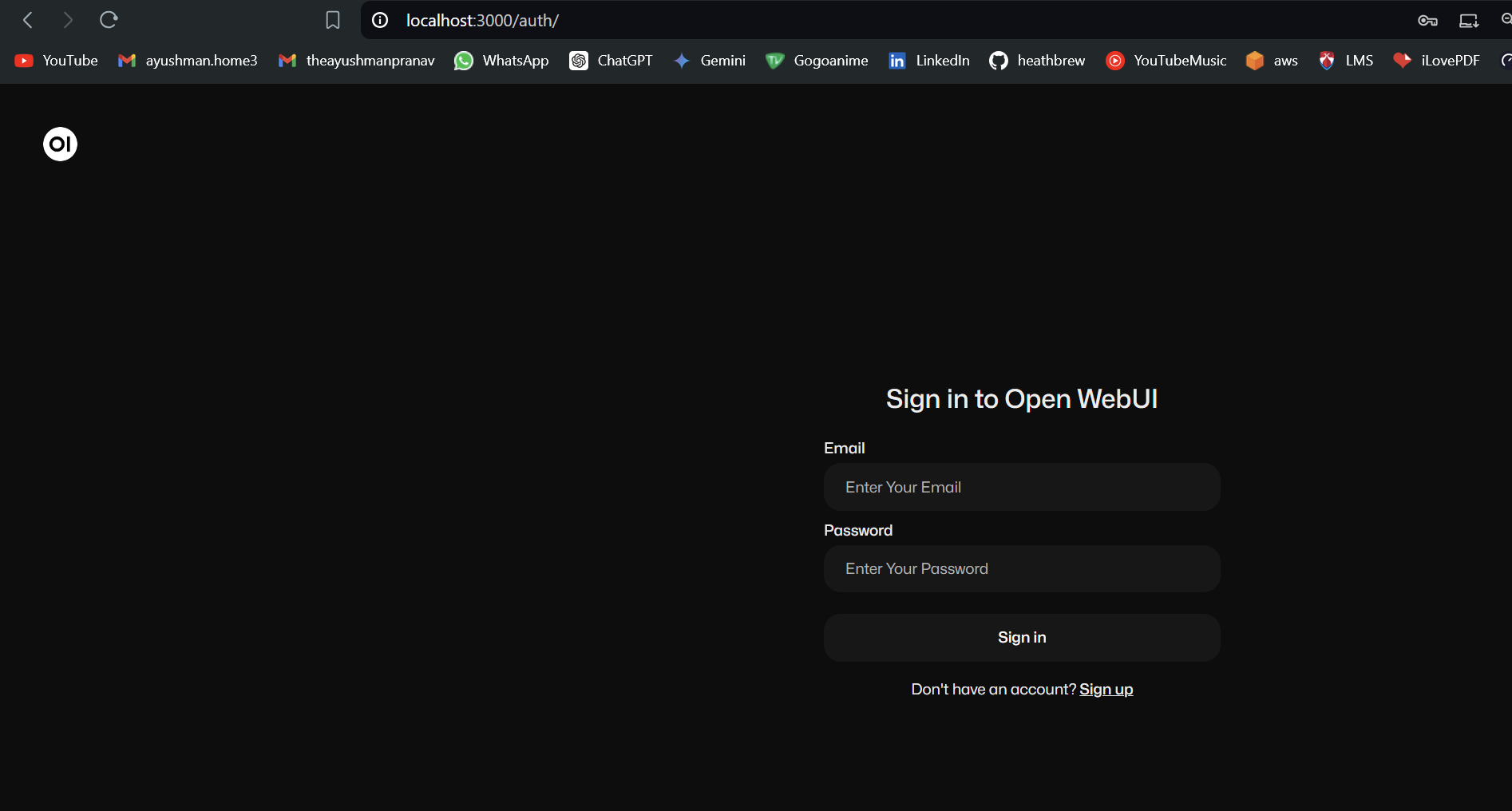
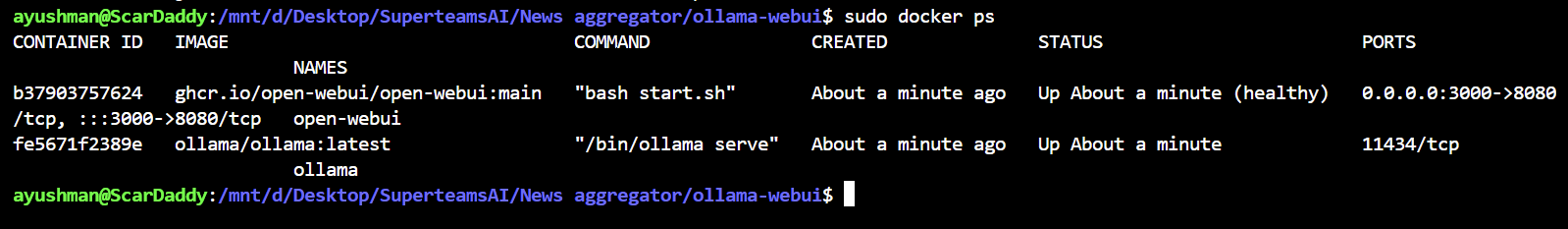
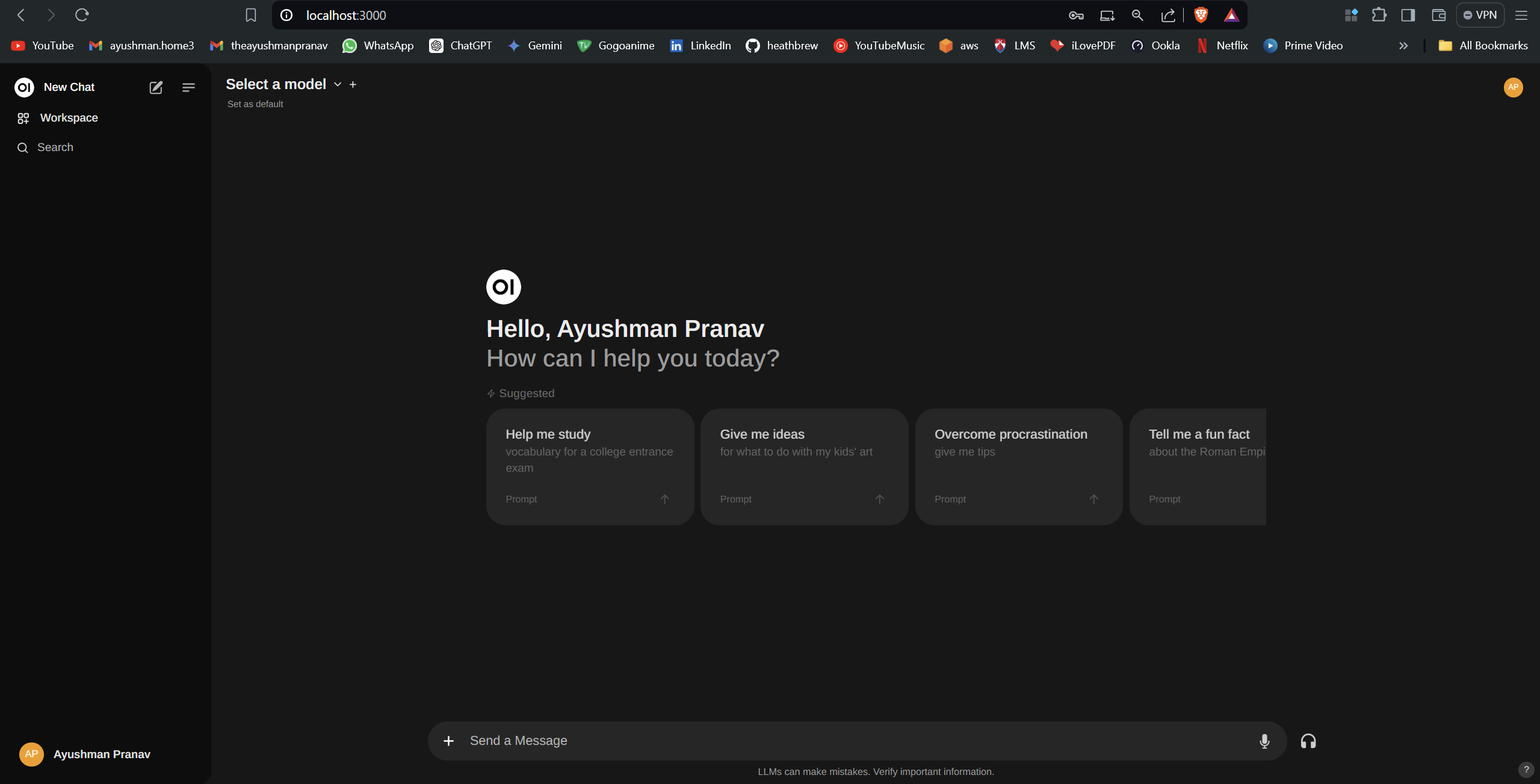
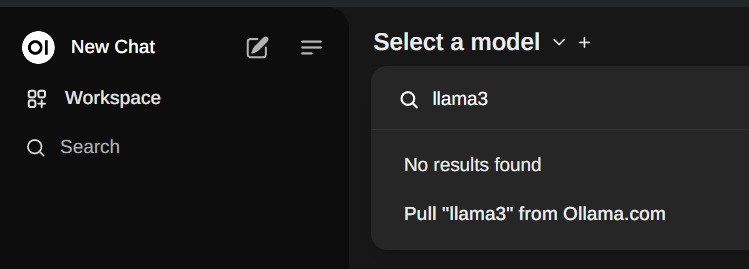
volumes:

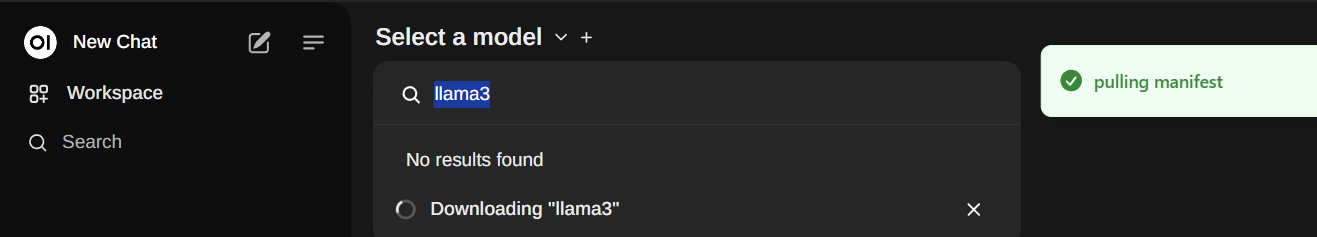
ollama: {}

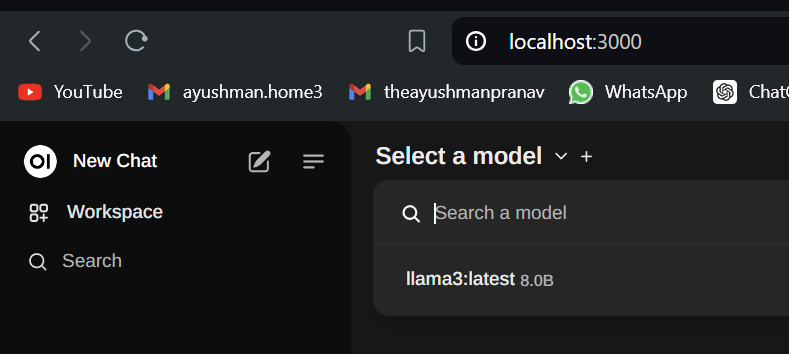
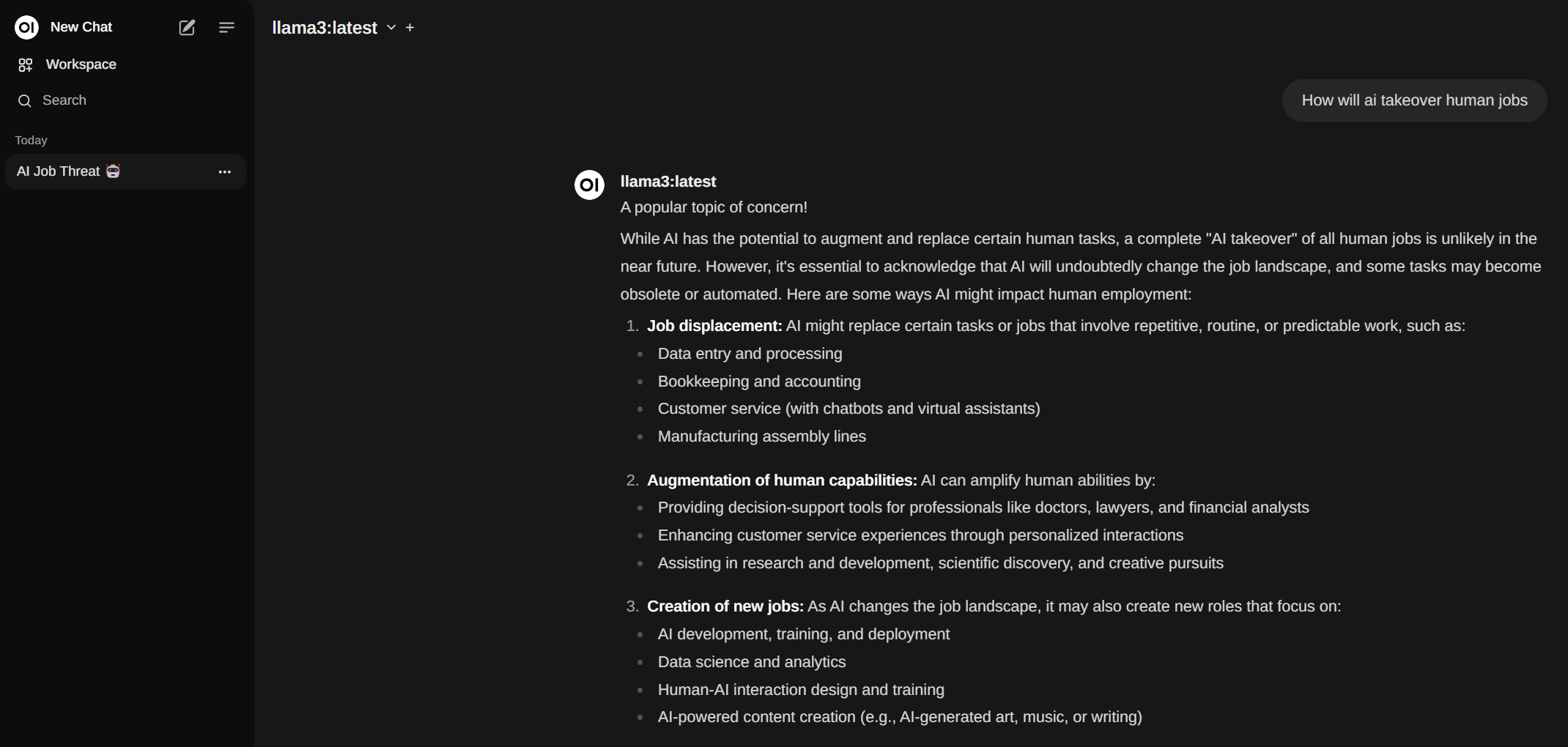
Ensure that you stop the Ollama Docker container before you run the following command:

docker compose up -d





Reload to get the functionality  
 

# Qdrant