**Docker Compose And Nagios**

**Docker Compose**

**1. Writing a Basic docker-compose.yml File**

**Step 1: Create a Folder**

Go to your desktop or any folder you prefer.

Right-click and select New Folder.

Name the folder my\_docker\_project.

**Step 2: Open a Text Editor**

Open Visual Studio Code, Notepad, or any other text editor.

Create a new file.

**Step 3: Write the YAML Configuration**

Here’s a simple example of a docker-compose.yml file for WordPress and MySQL:

version: '3.1'

services:

db:

2 RKR21 SOFTWARE ENGINEERING LAB CSE/IT/CSM/CSD III/I

image: mysql:5.7

container\_name: mysql\_container

environment:

MYSQL\_ROOT\_PASSWORD: rootpassword

MYSQL\_DATABASE: wordpress\_db

MYSQL\_USER: wordpress\_user

MYSQL\_PASSWORD: wordpress\_pass

volumes:

- db\_data:/var/lib/mysql

wordpress:

depends\_on:

- db

image: wordpress:latest

container\_name: wordpress\_container

ports:

- "8000:80"

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress\_user

WORDPRESS\_DB\_PASSWORD: wordpress\_pass

WORDPRESS\_DB\_NAME: wordpress\_db

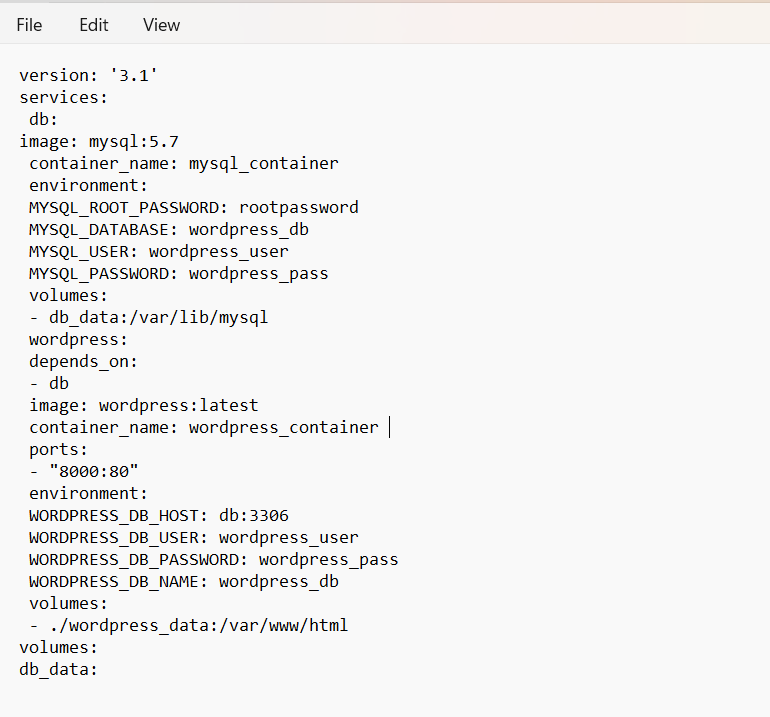
volumes:

- ./wordpress\_data:/var/www/html

volumes:

3 RKR21 SOFTWARE ENGINEERING LAB CSE/IT/CSM/CSD III/I

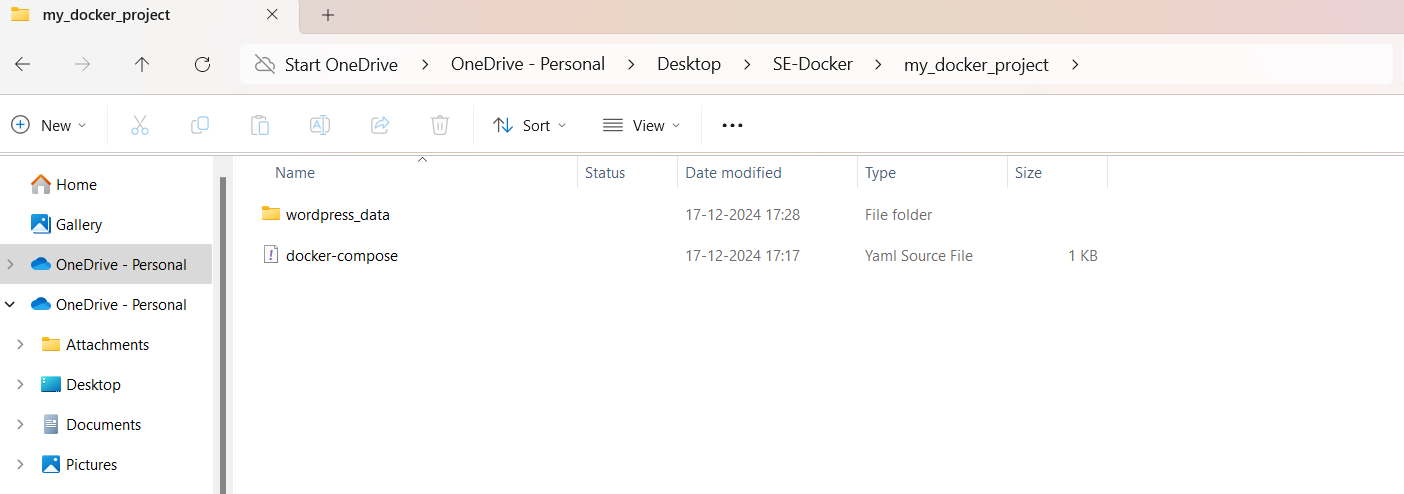
db\_data:



**2. Saving the File**

Save the file as docker-compose.yml.

Place it in the my\_docker\_project folder.



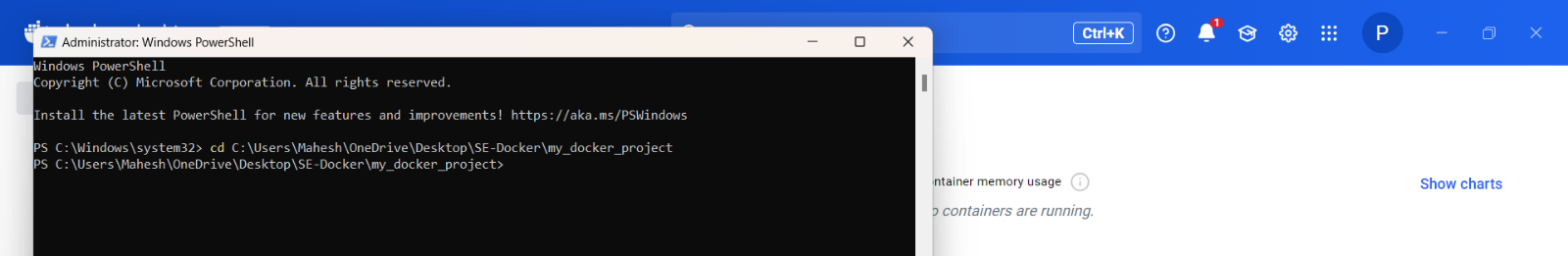
**3. Running the Setup**

**Step 1: Open Command Line**

Open PowerShell or Command Prompt.

Navigate to the my\_docker\_project folder:

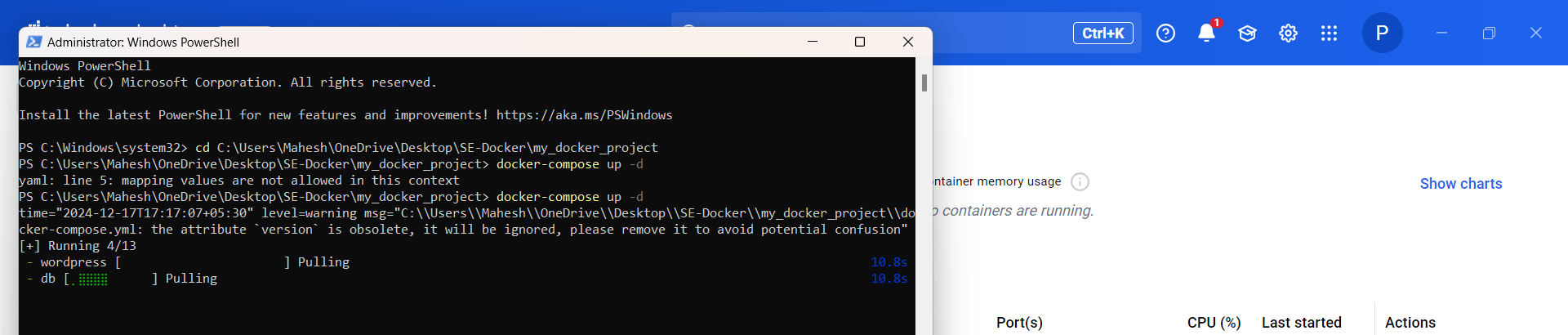
cd path\_to\_my\_docker\_project



**Step 2: Start the Containers**

Run:

**docker-compose up -d**



This command reads the docker-compose.yml file and creates both the WordPress and MySQL

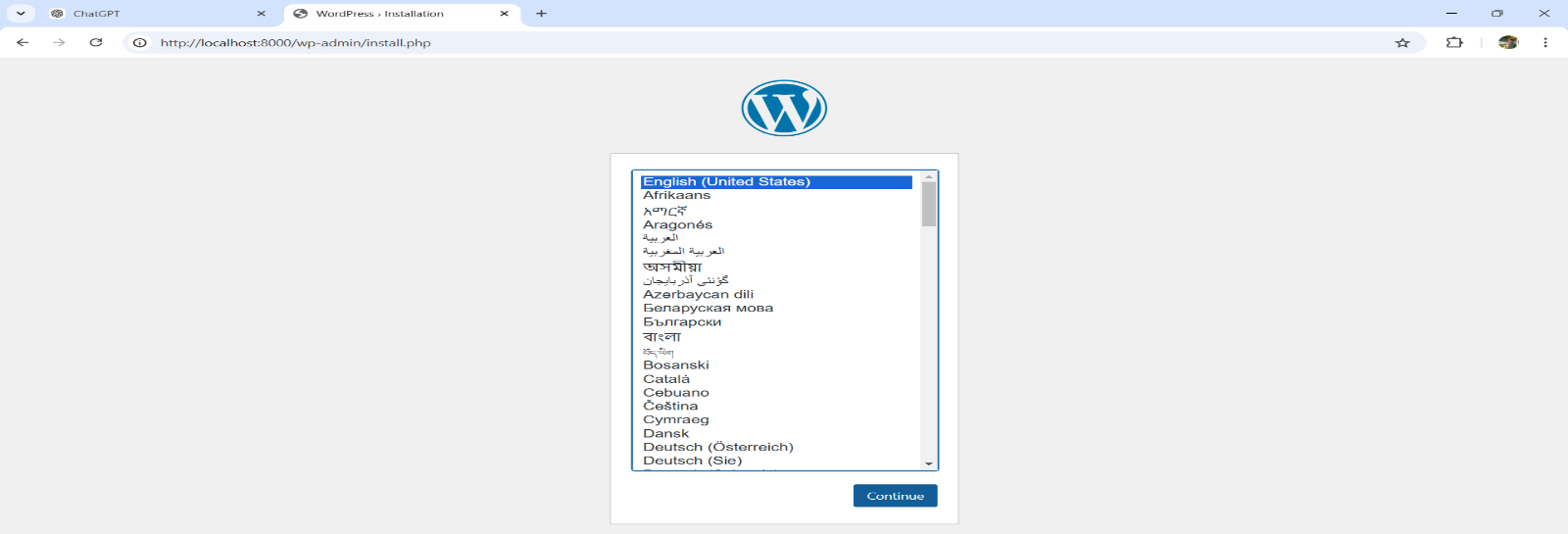
containers.

-d runs the containers in the background.

**4. Accessing the Application**

Open your web browser.

Go to [**http://localhost:8000**](http://localhost:8000)

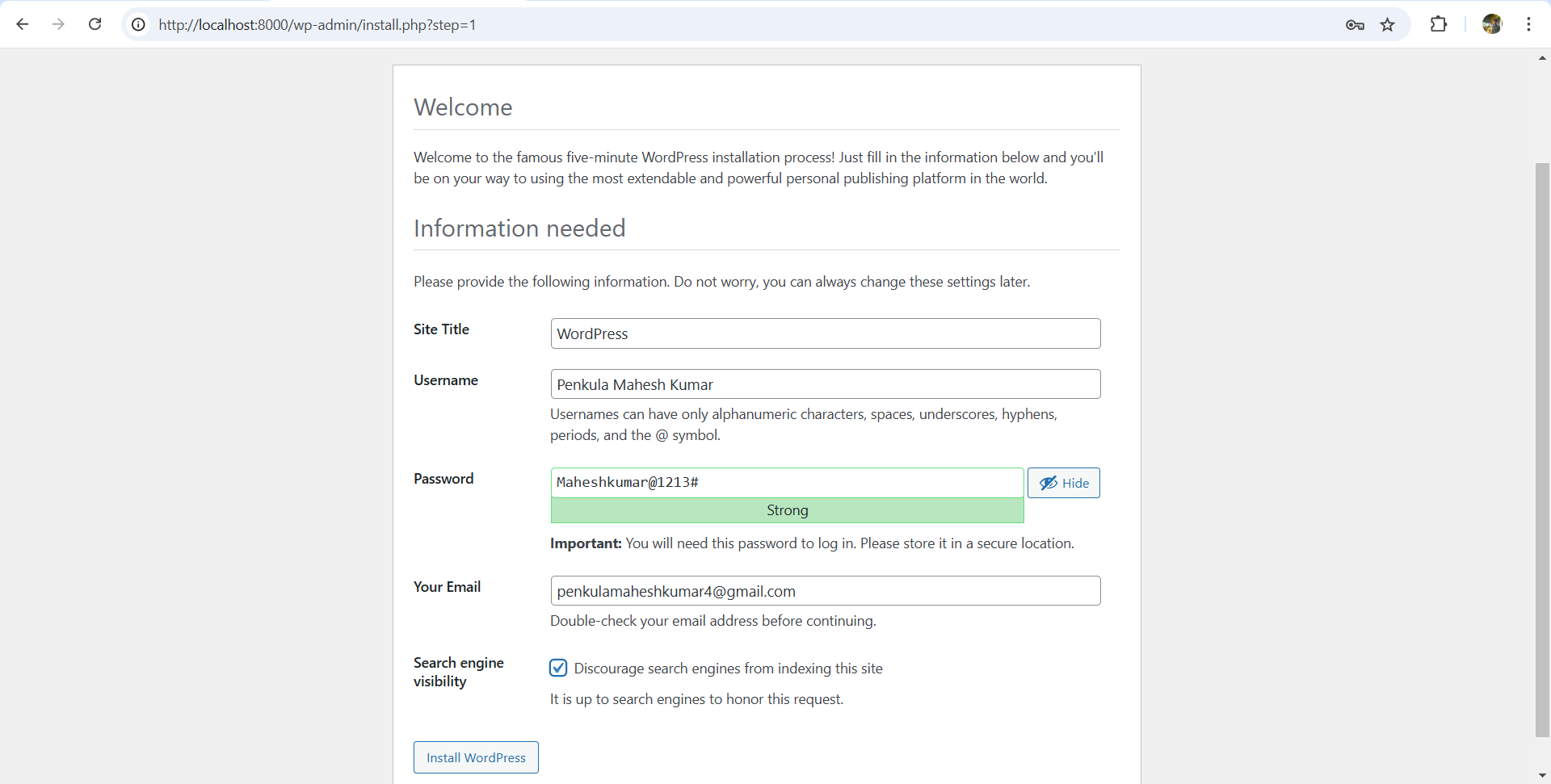


Follow the WordPress setup wizard to complete the installation:

Site Name.

Admin Username and Password.

Email Address.



**8. Managing Containers**

Stop the Containers

To stop the containers without removing them:

**docker-compose stop**

Start Again

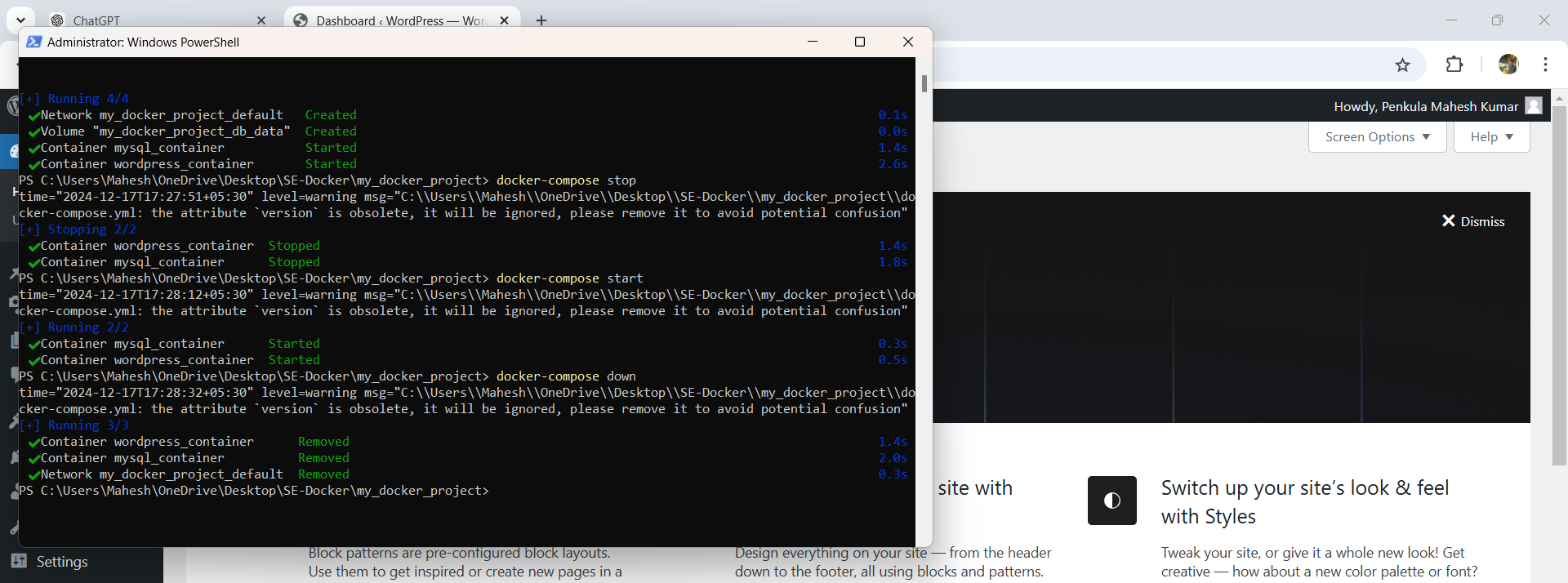
To restart the containers:

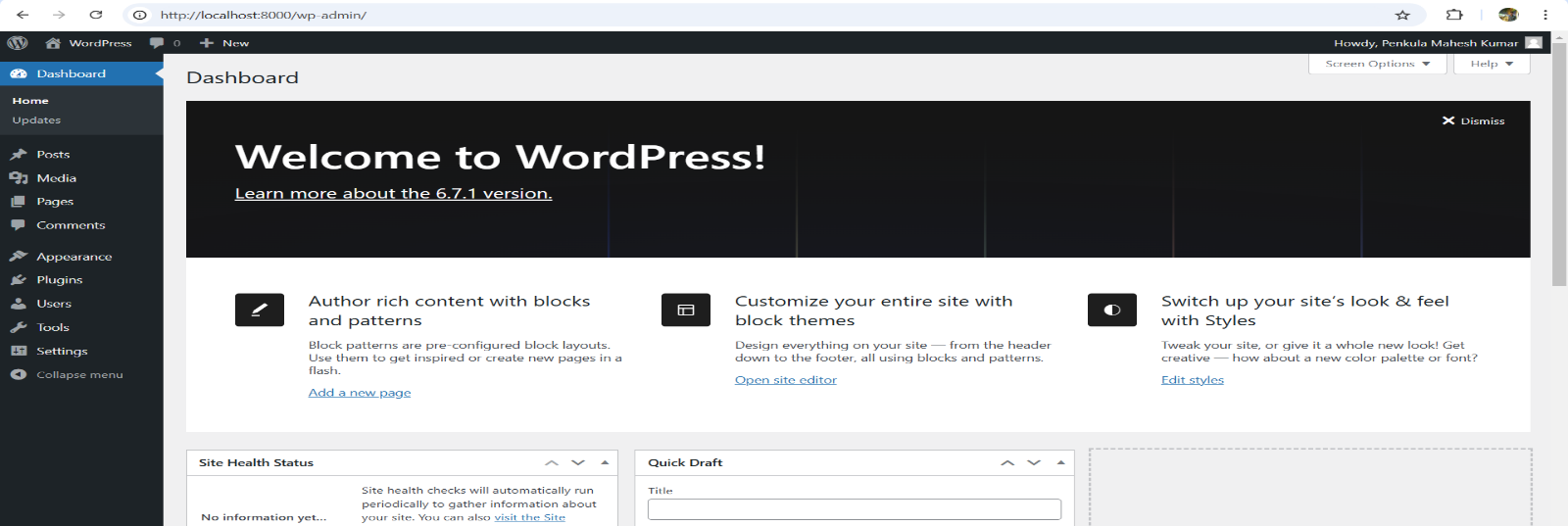
**docker-compose start**

Remove Containers

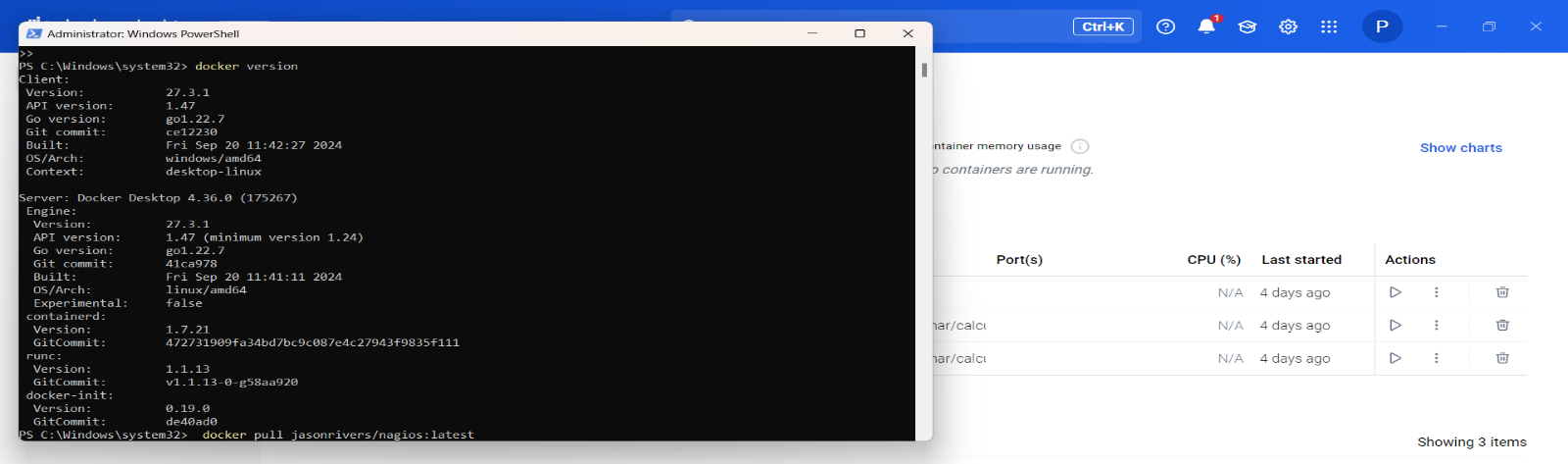
To stop and remove everything:

**docker-compose down**



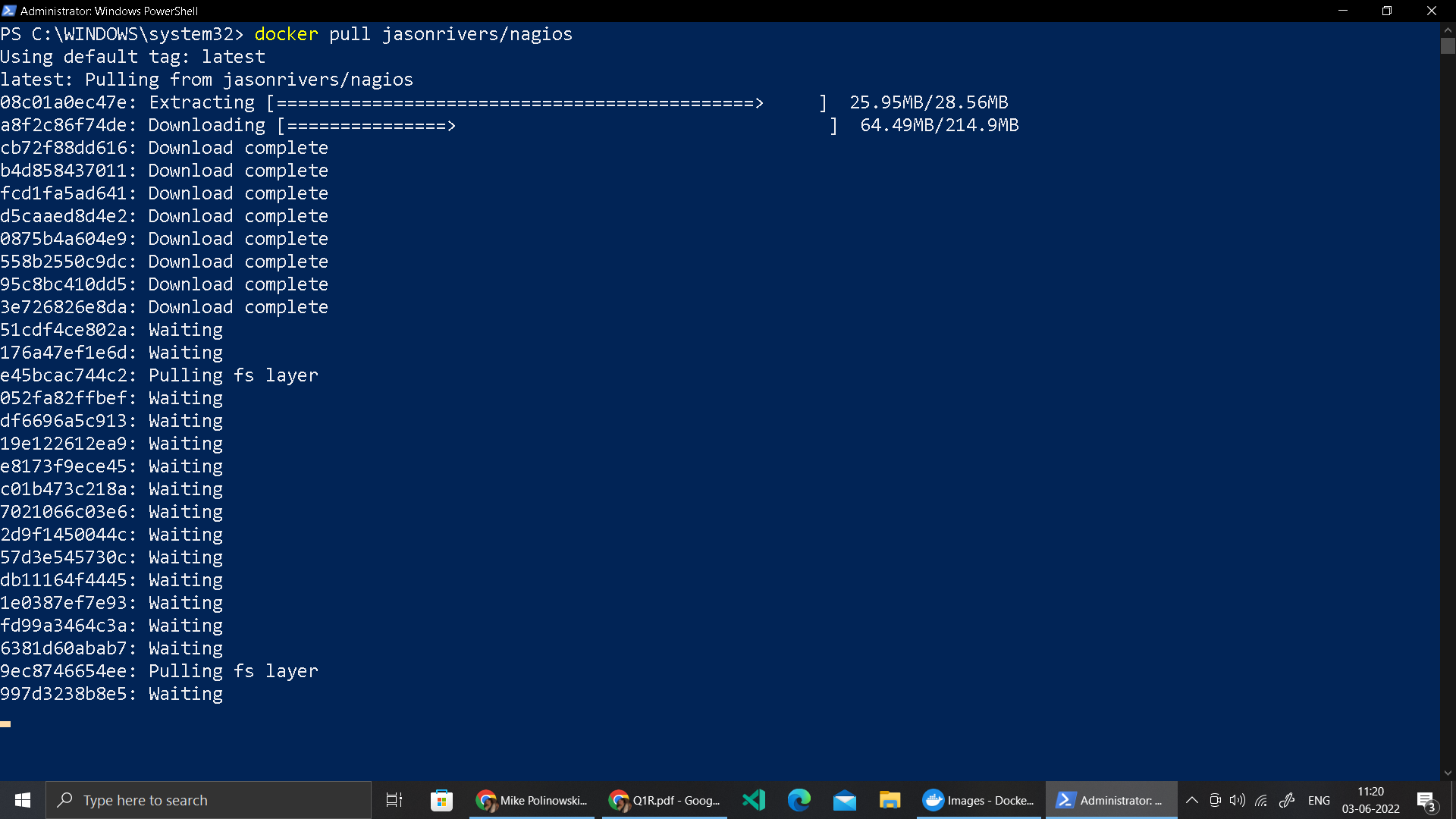


**Nagios**

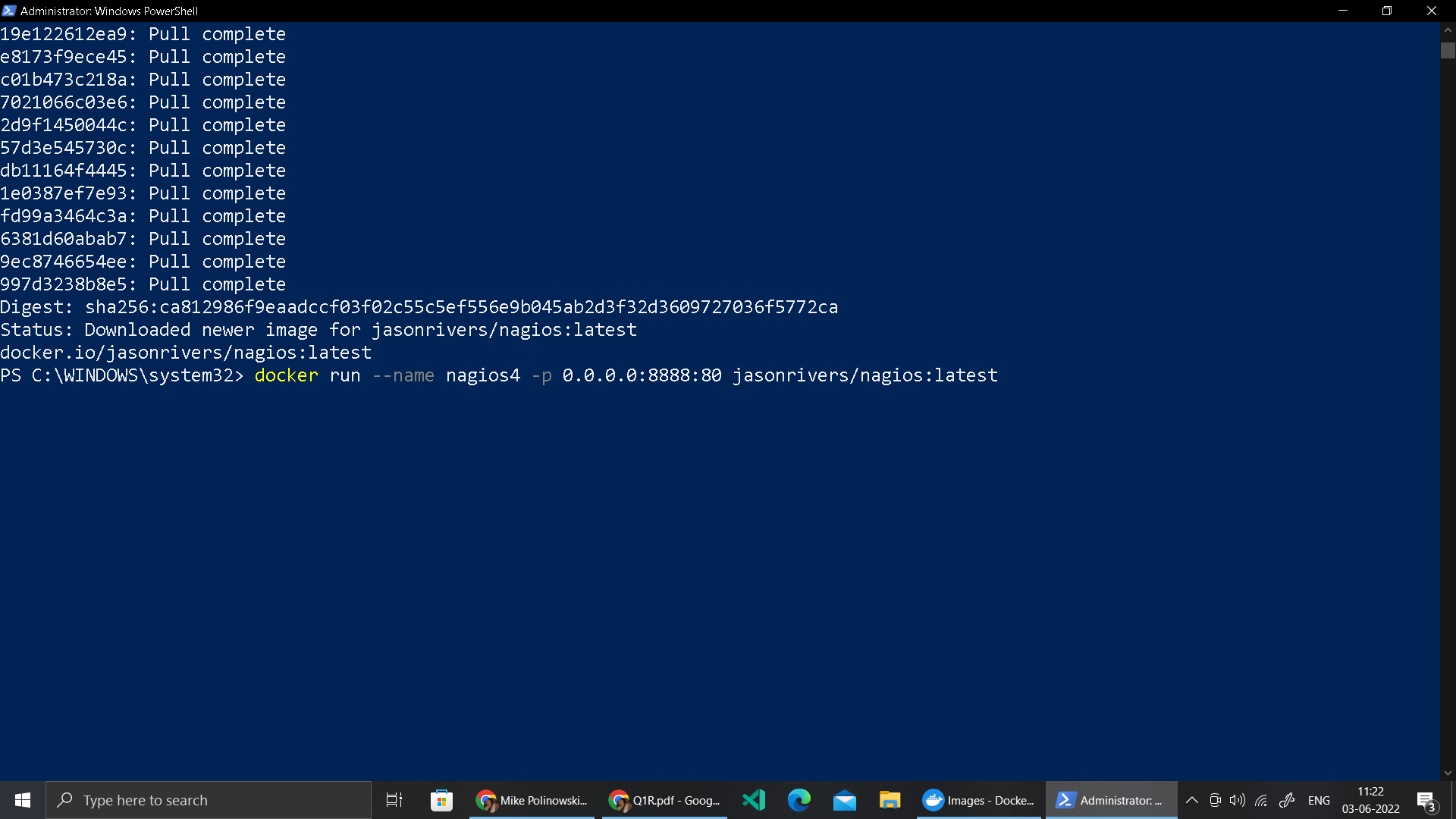
1.Run Docker Desktop 2.Open PowerShell as administrator and type the following command **docker version** [to check that both client and dockere daemon are running] 

Pull the Nagios image from the Docker Hub with the following command

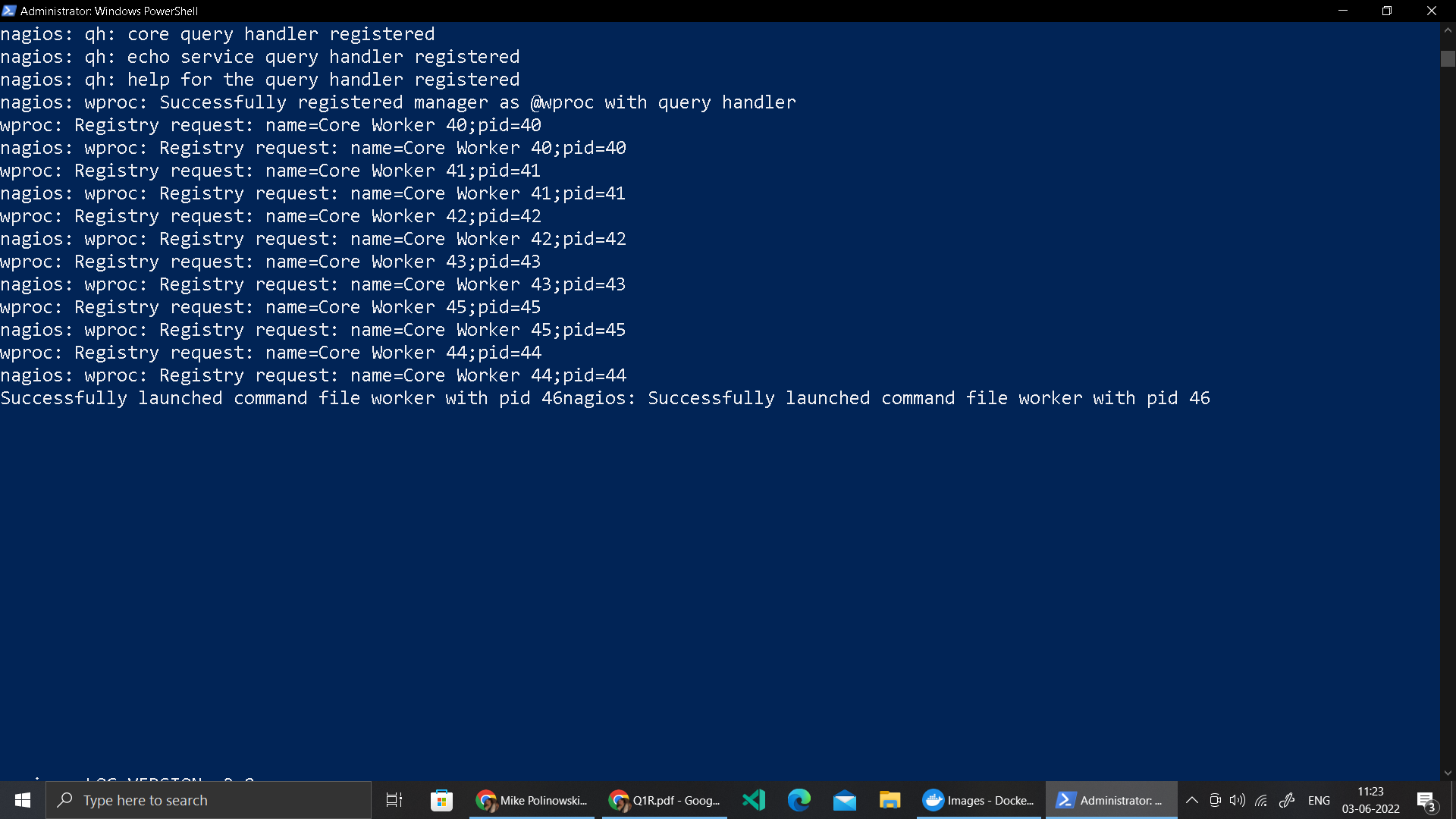
**docker pull jasonrivers/nagios:latest**



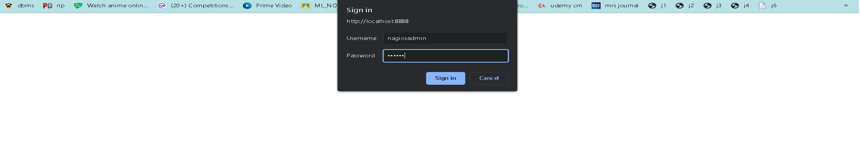
**docker run --name nagios4 -p 0.0.0.0:8888:80 jasonrivers/nagios:latest**



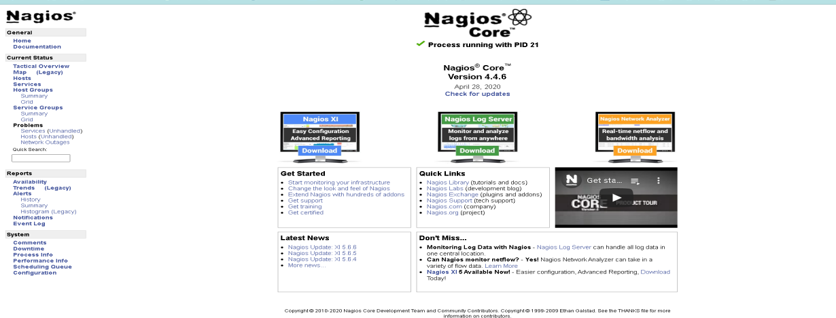
Nagios is successfully launched



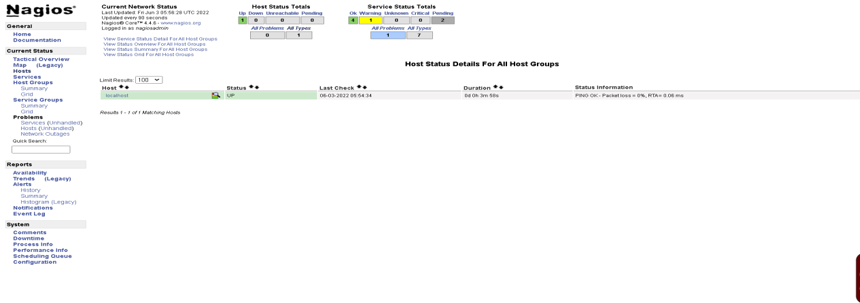
We can access Nagios at the browser at port **8888** [**localhost:8888**]The default credentials for the web interface is nagiosadmin / nagios



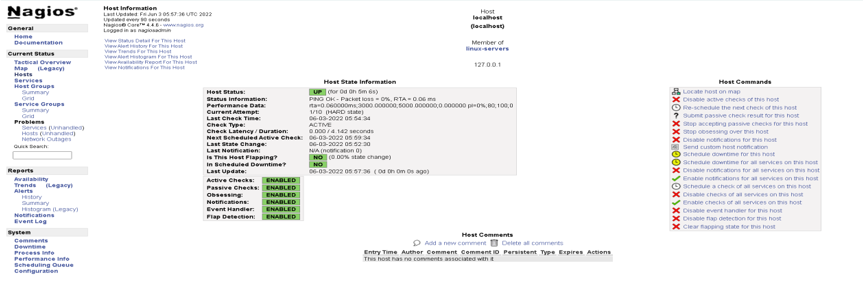
Nagios dashboard is the interface to monitor the services



Host services [ left side pane can be used for selection.

Click on localhost to check the Host State information

Click on localhost to check the Host State information



Click on services to check the various services running **CurrentLoad, Current User, HTTP, PING etc**

