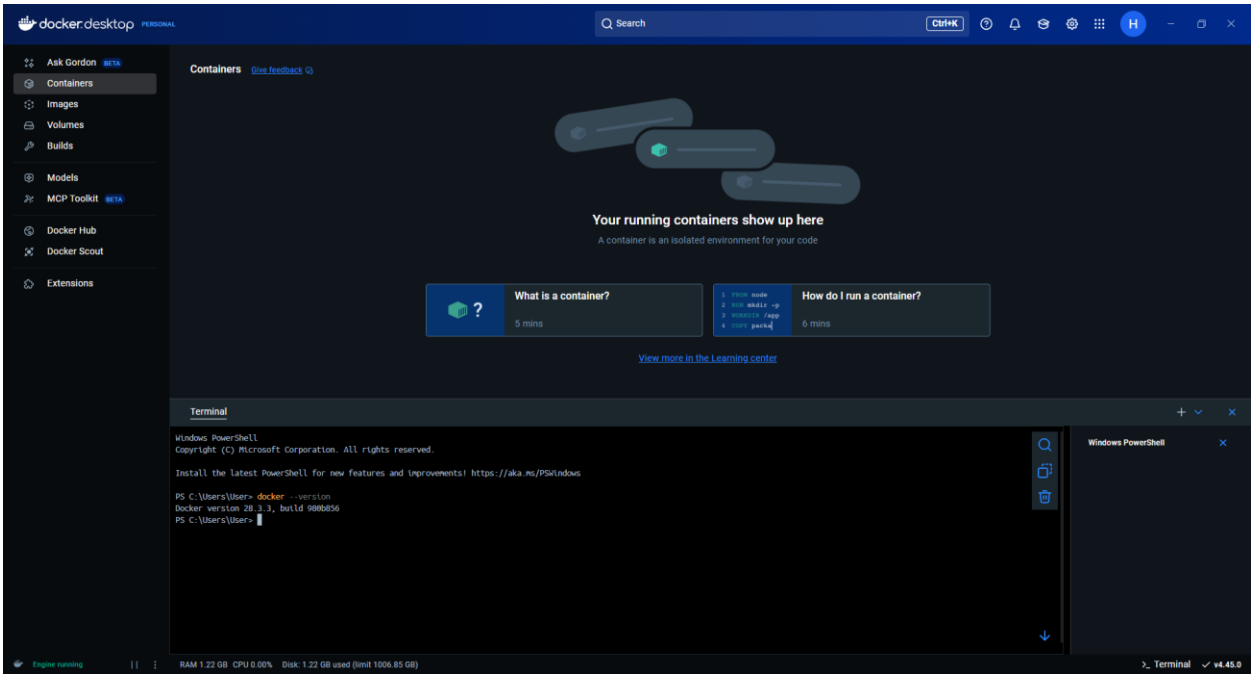
Nguyễn Việt Hoàng

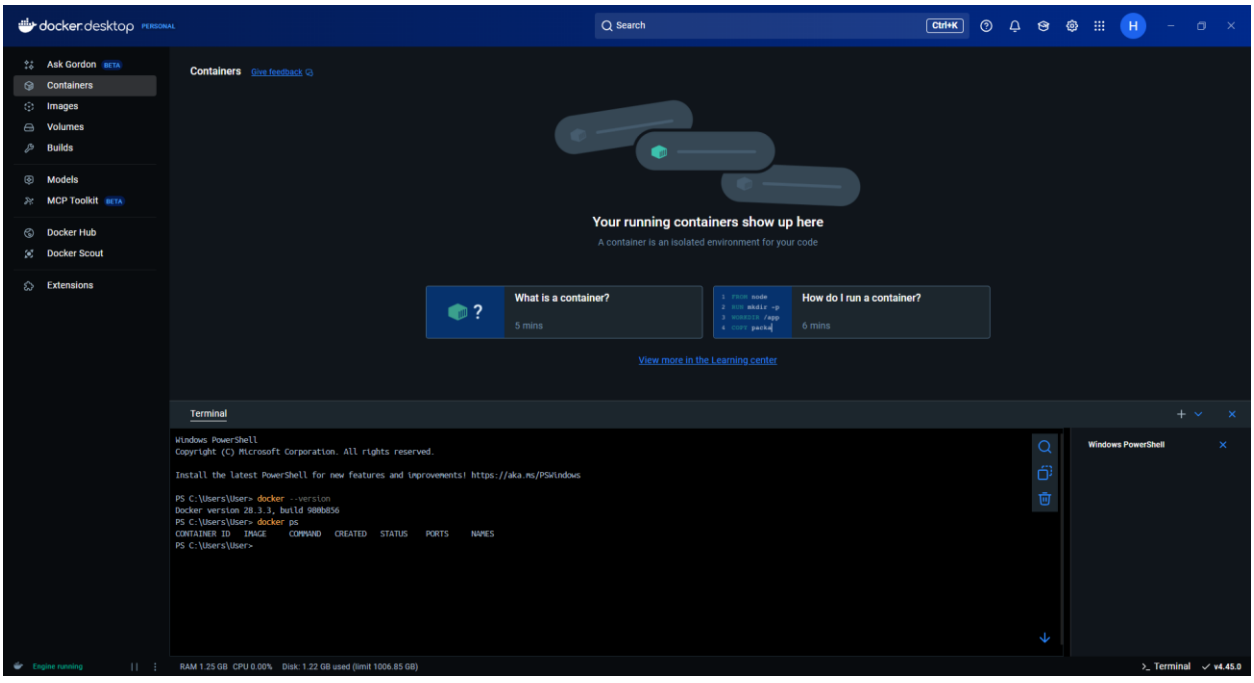
B22DCVT214

1. Docker Basic Commands

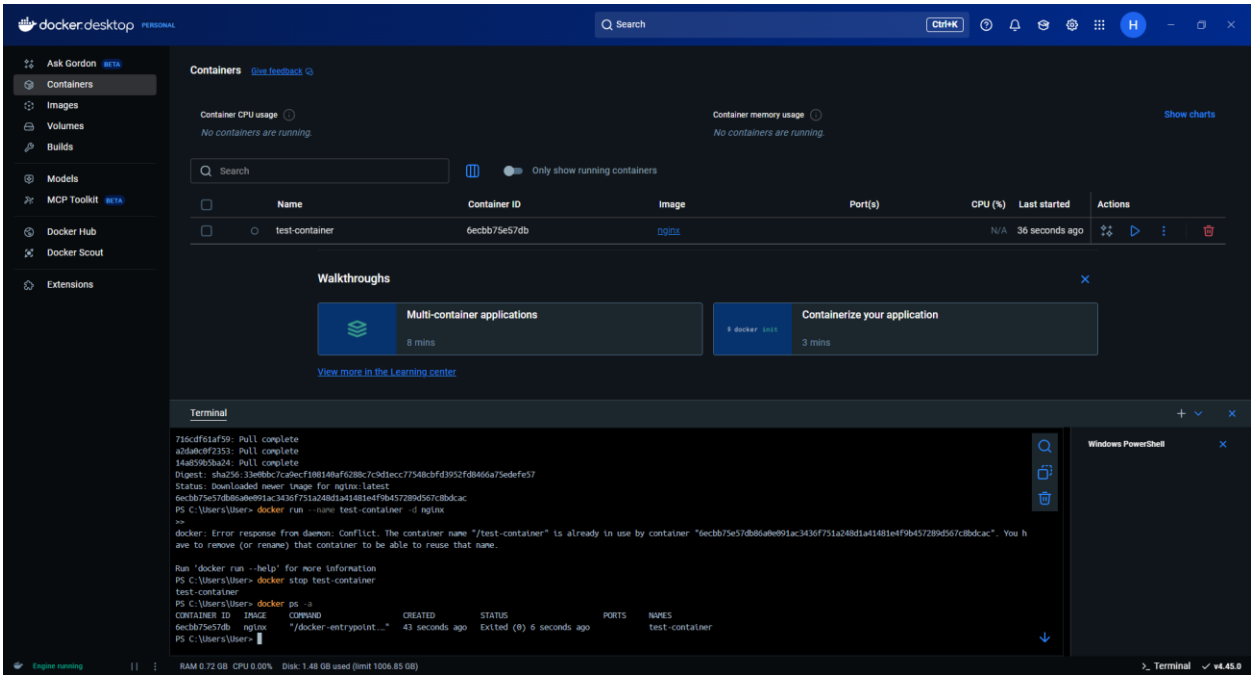
Check Docker version with docker --version



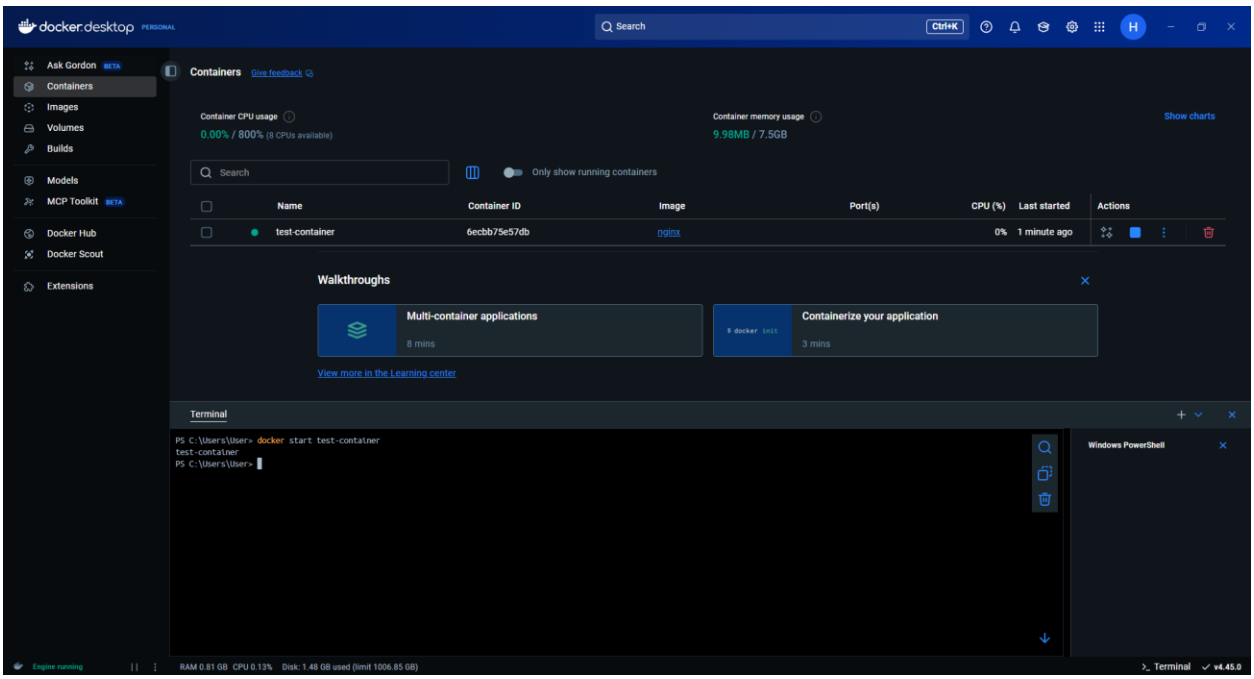
List running containers with docker ps



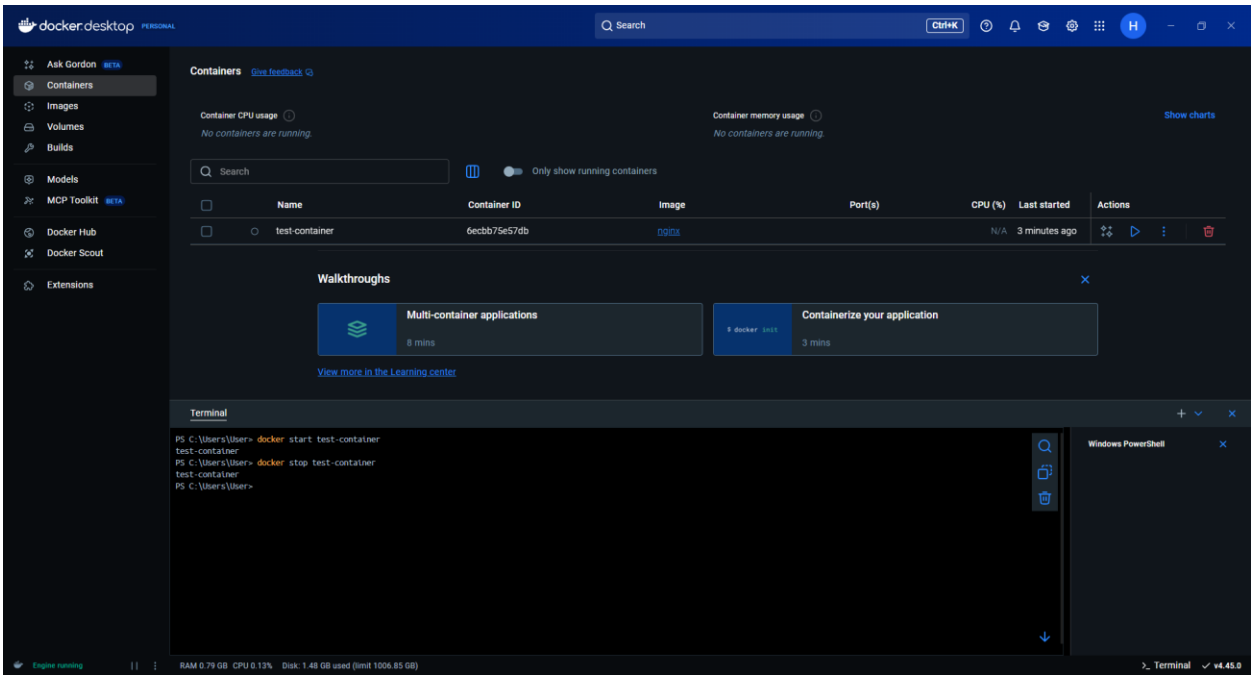
List all containers (including stopped) with docker ps -a



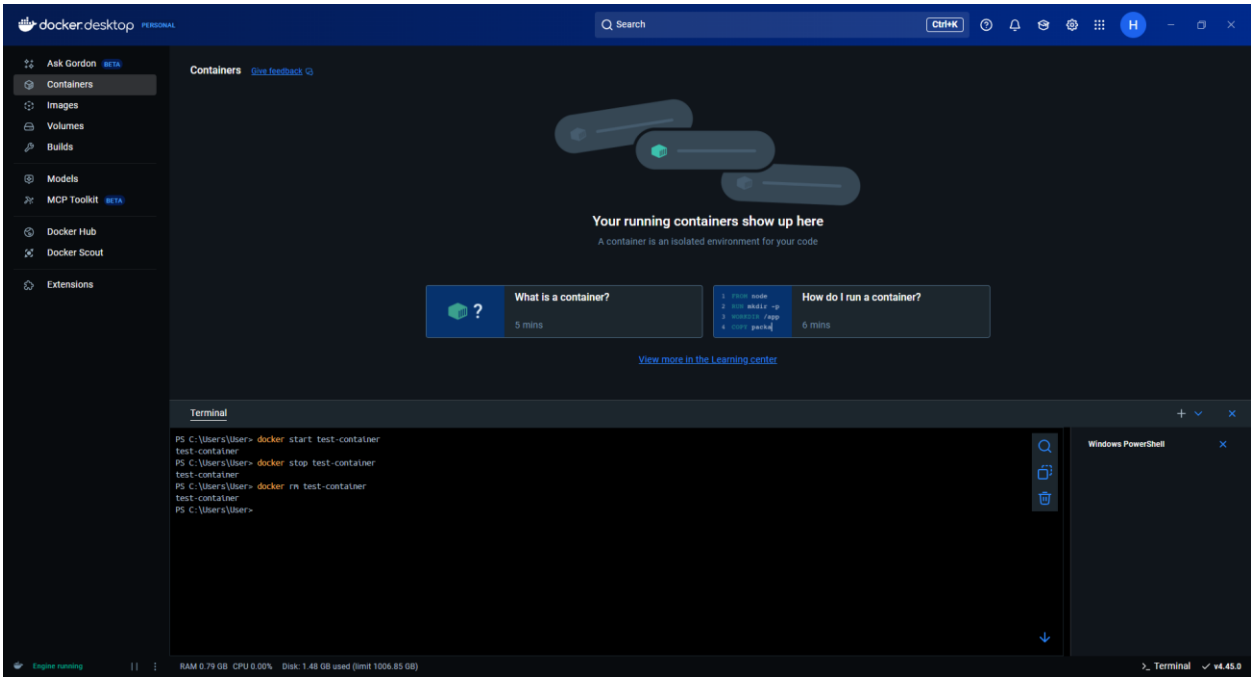
Start a container



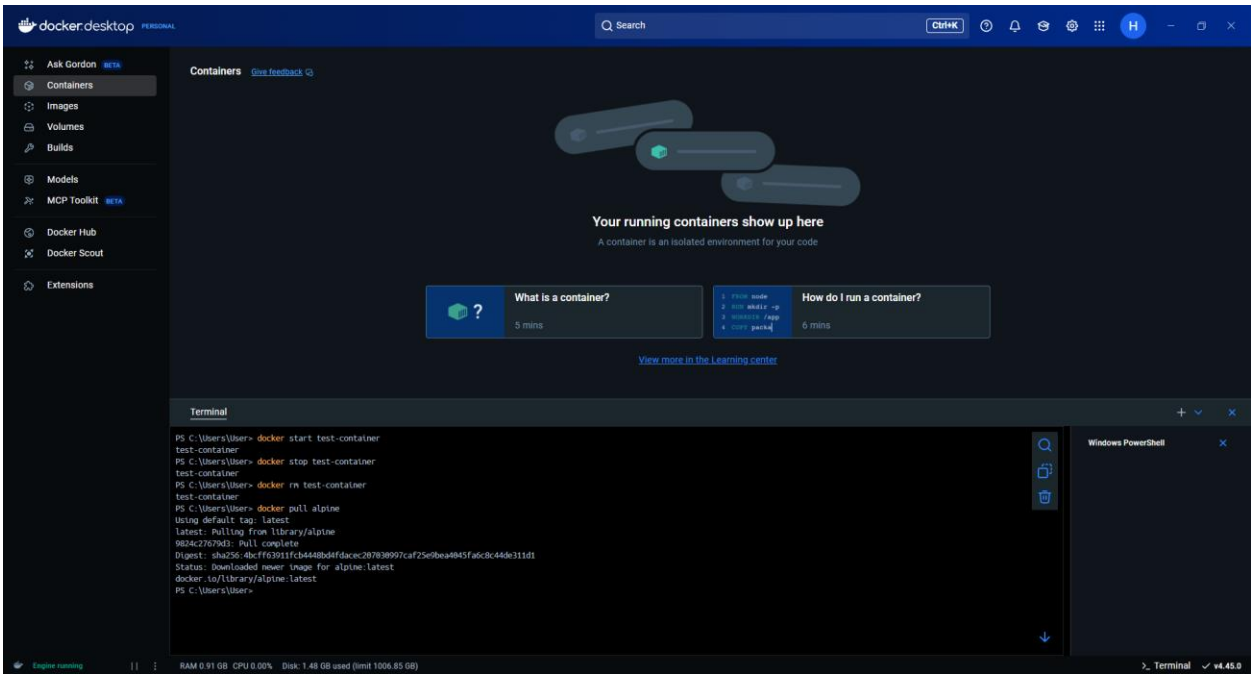
Stop a container



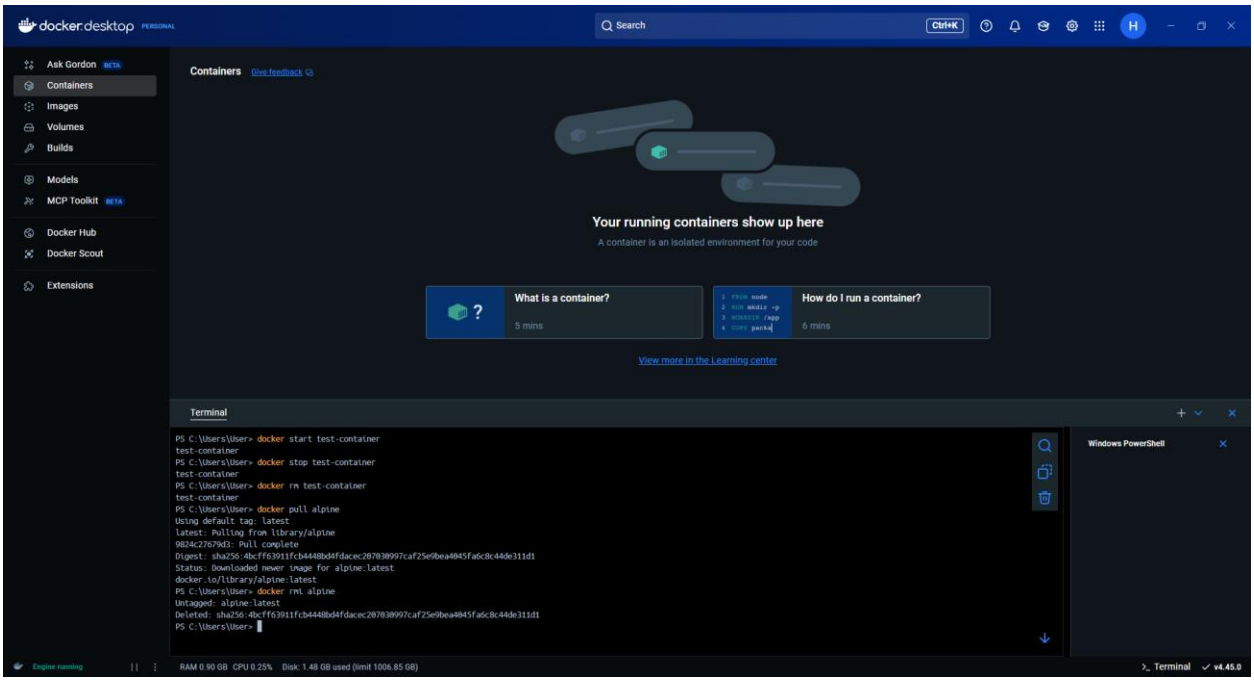
Remove a container



Pull an image

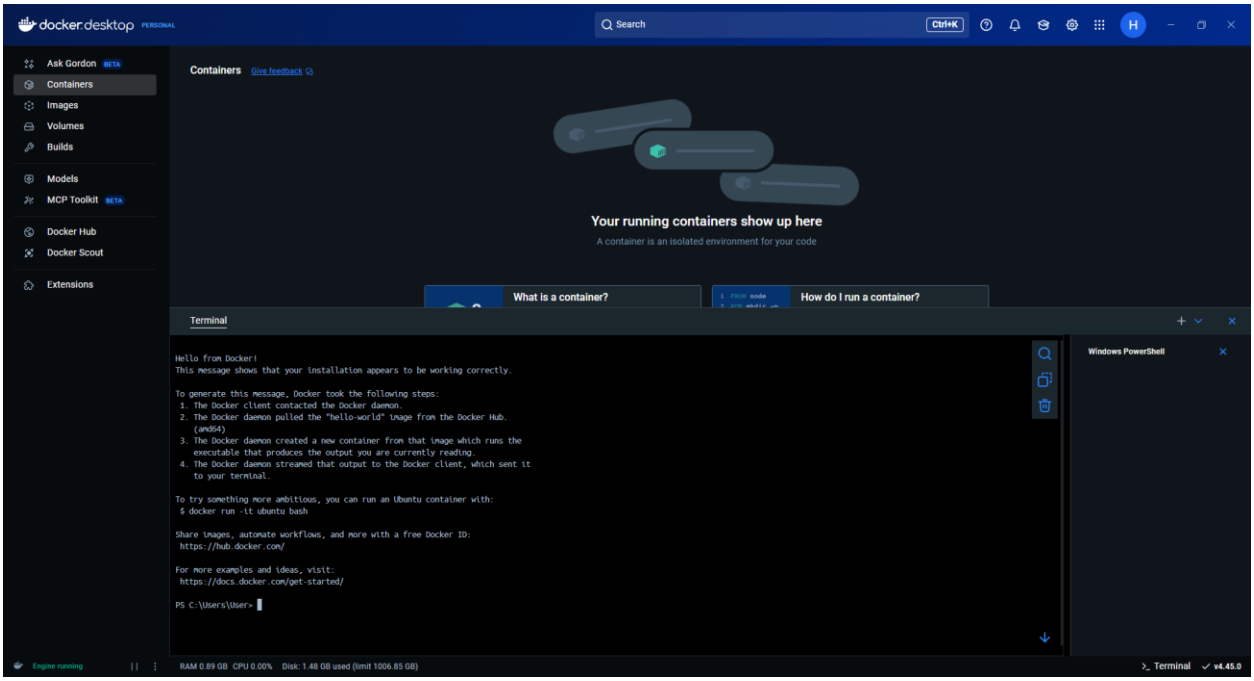


Remove an image with docker rmi

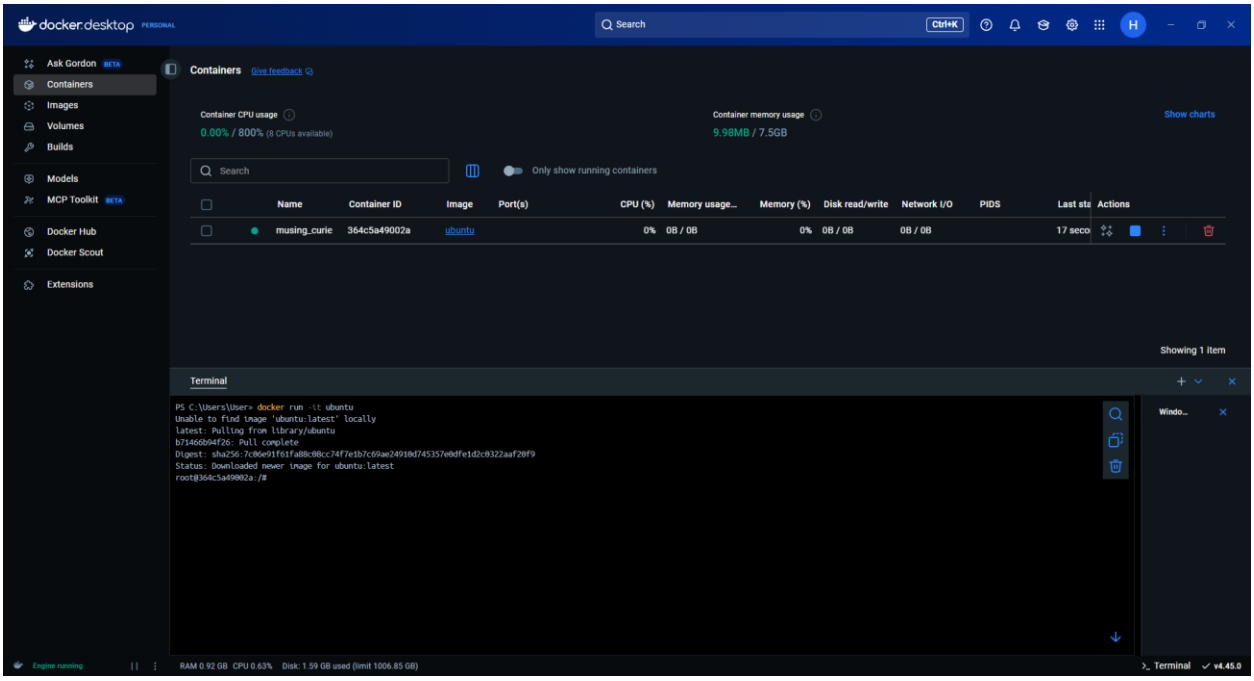


2. Docker Run

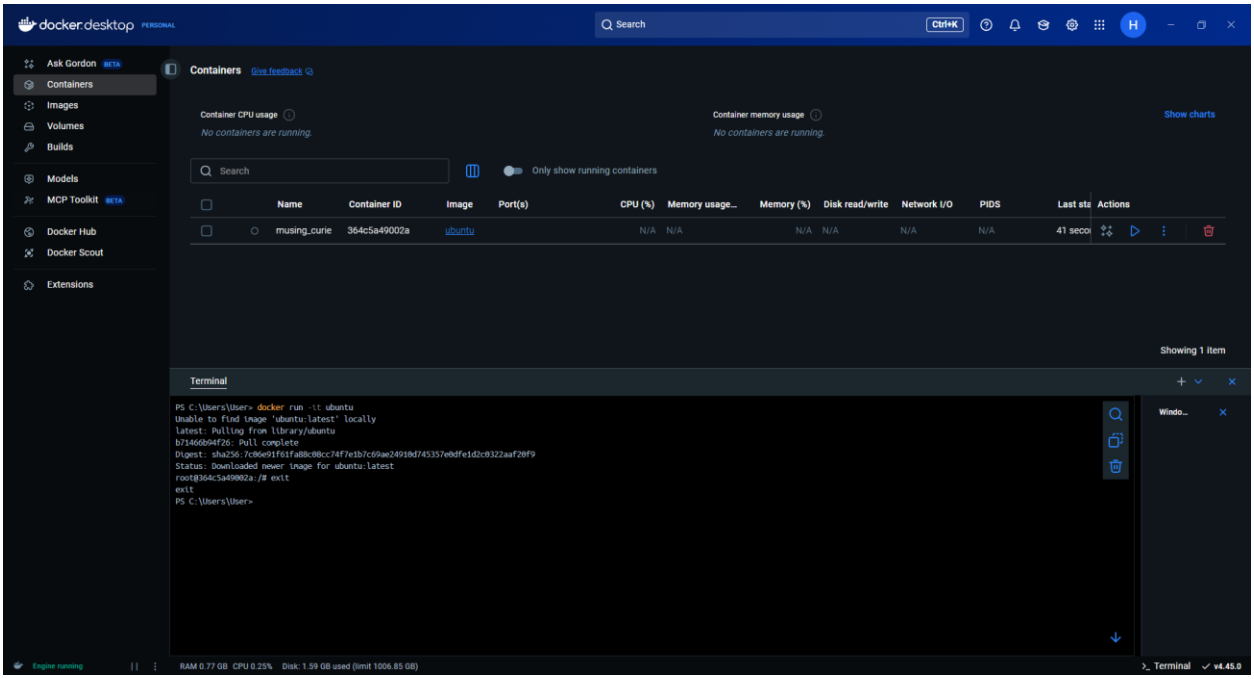
Run the hello-world container



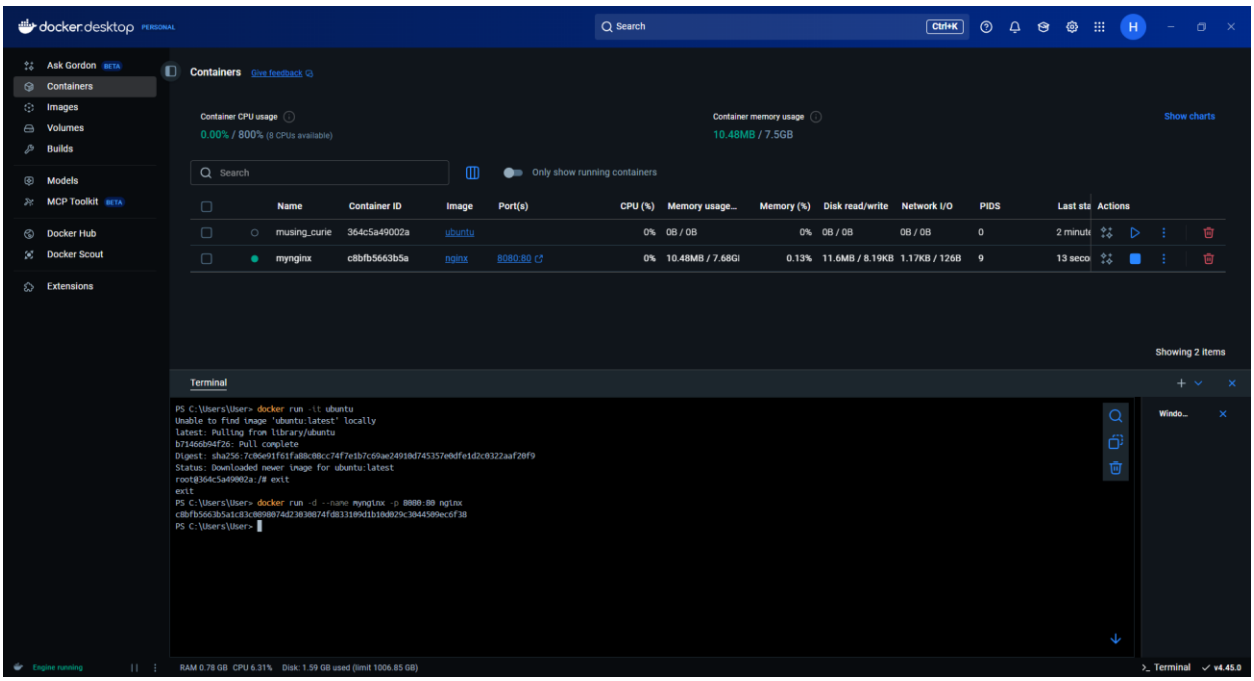
Run an Ubuntu container in interactive mode



Exit Ubuntu container

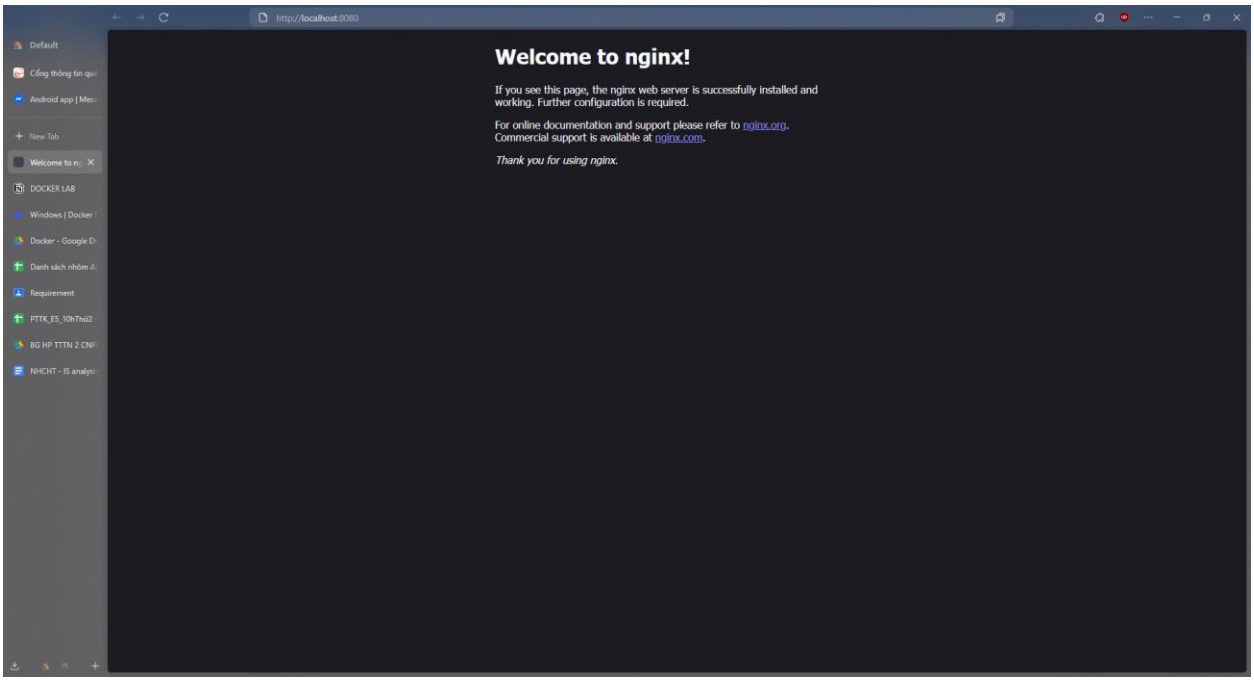


Run an Nginx container mapping host port 8080→ container port 80



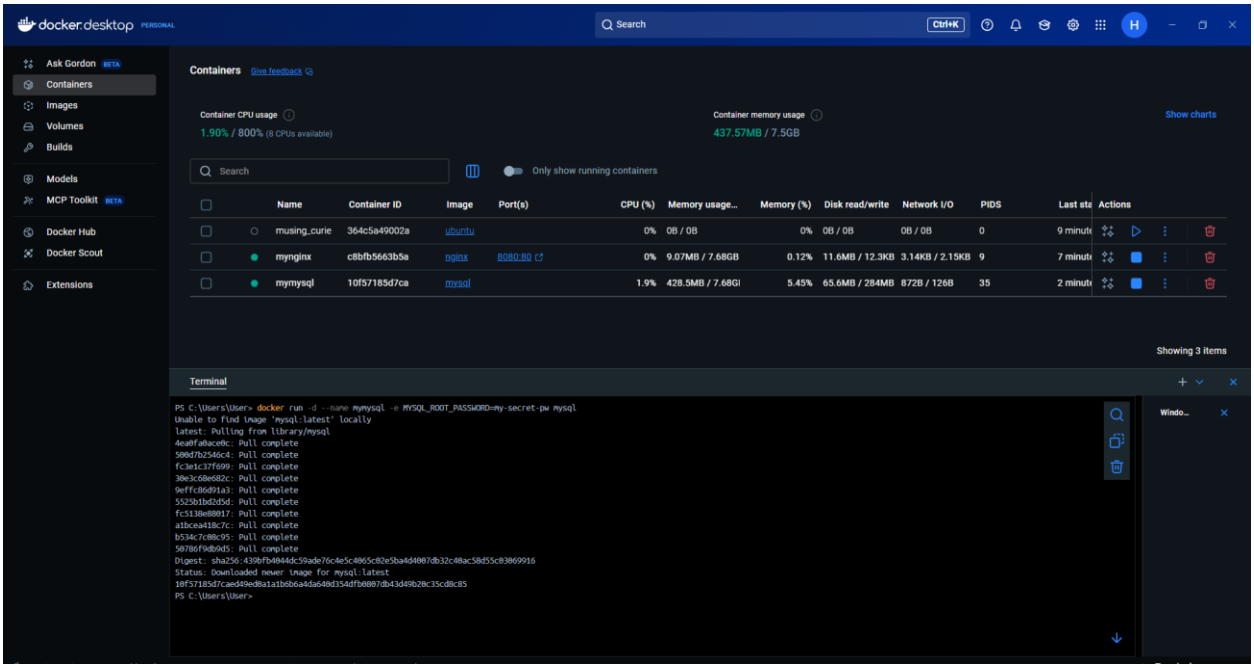
Access Nginx via browser at http://localhost:8080

.

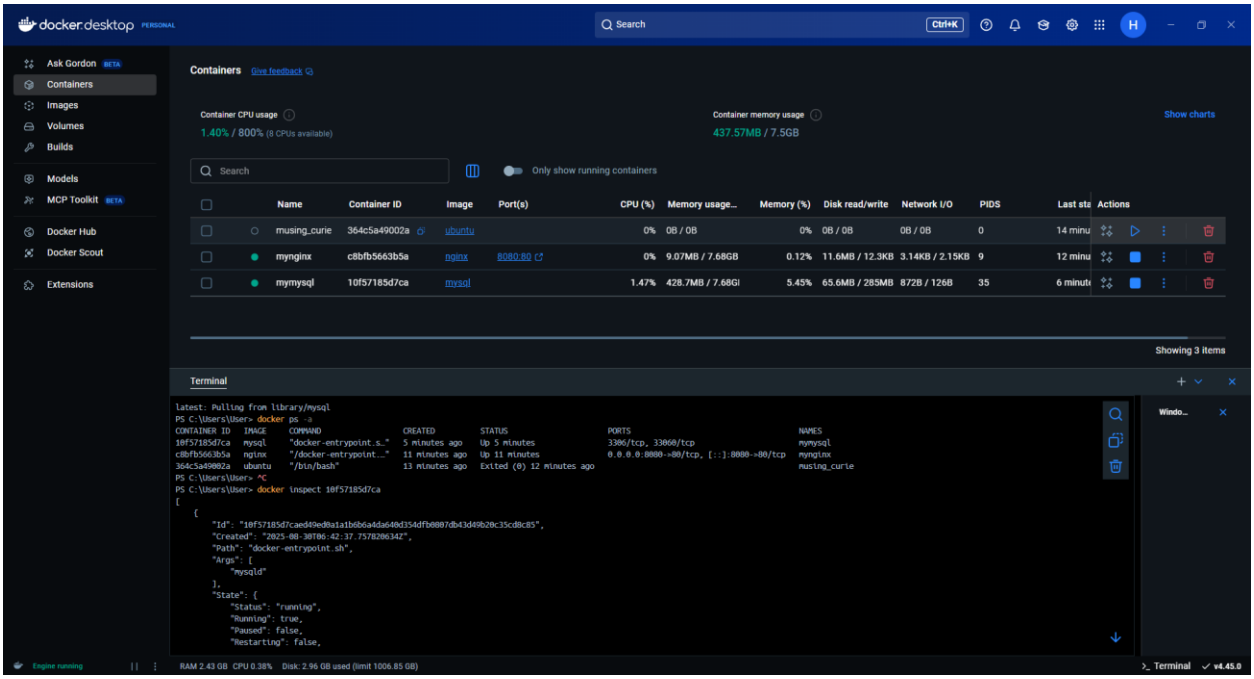


3. Docker Environment Variables

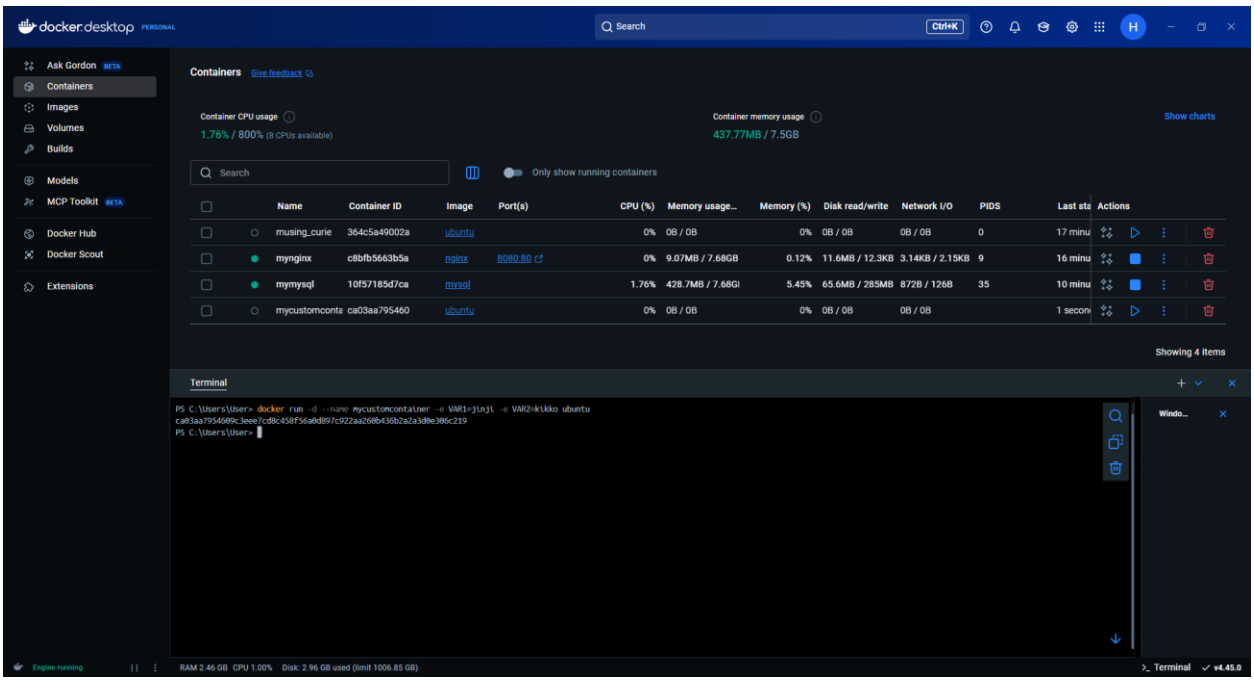
Run MySQL container with e MYSQL_ROOT_PASSWORD=my-secret-pw



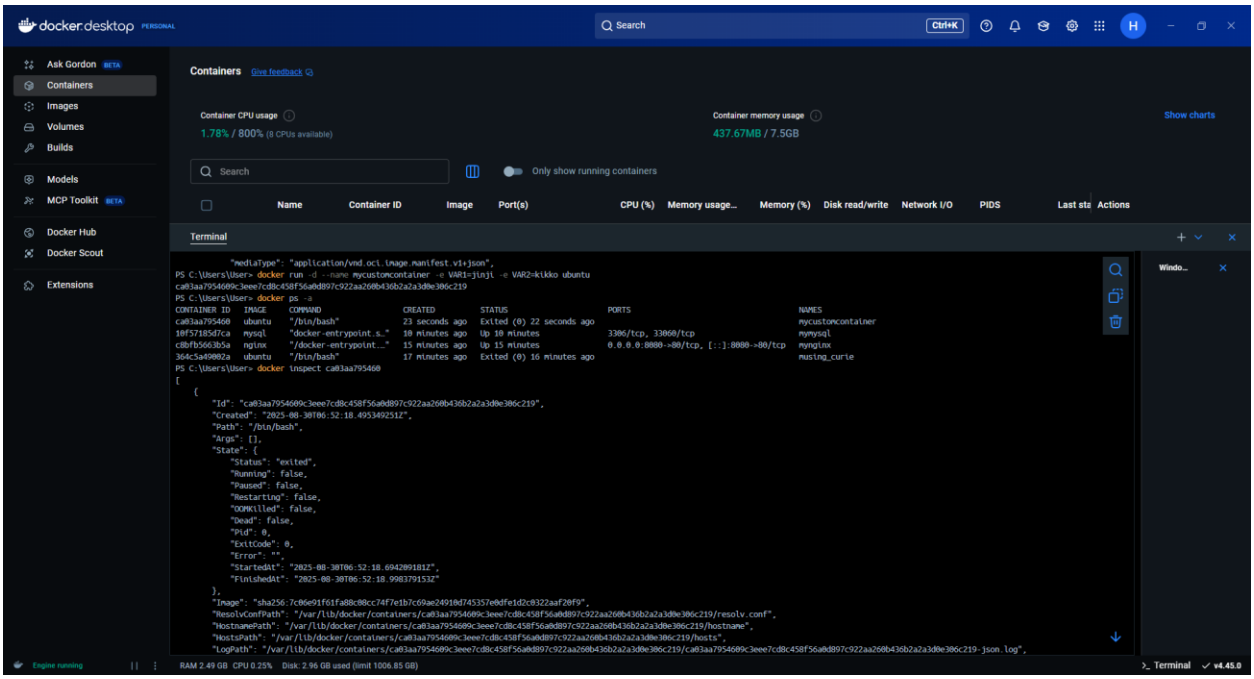
Verify variables with docker inspect <container_id>



Run a custom container with user-defined env vars.

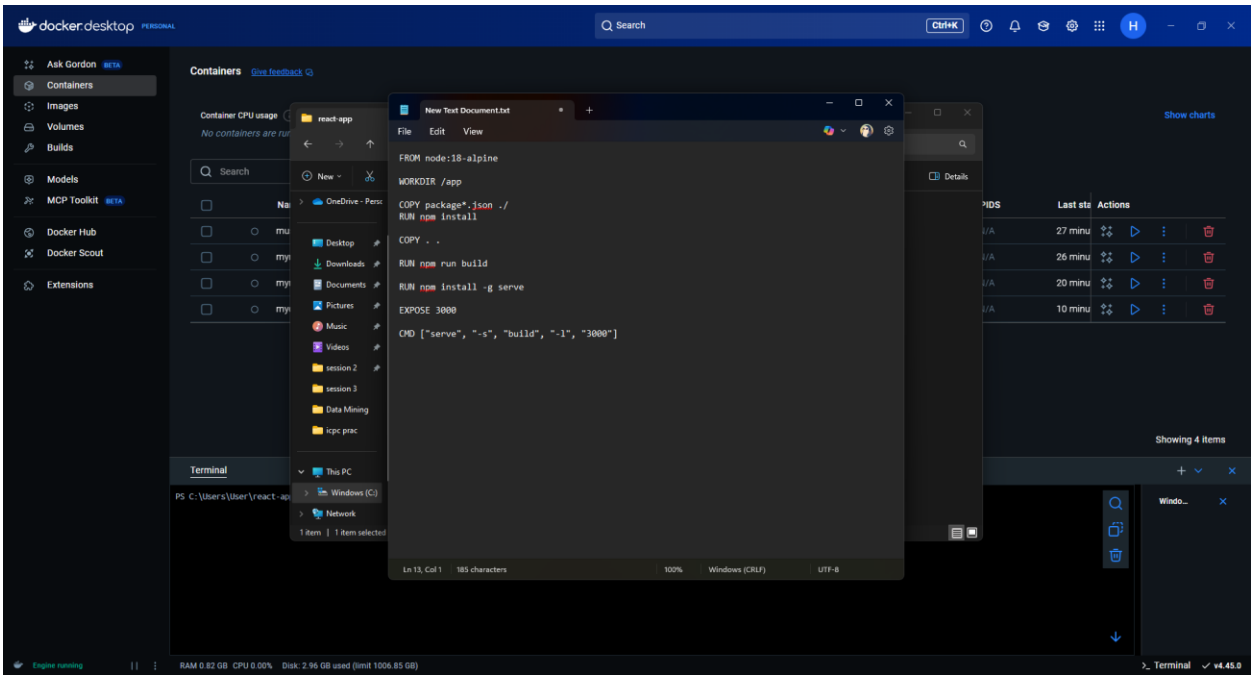


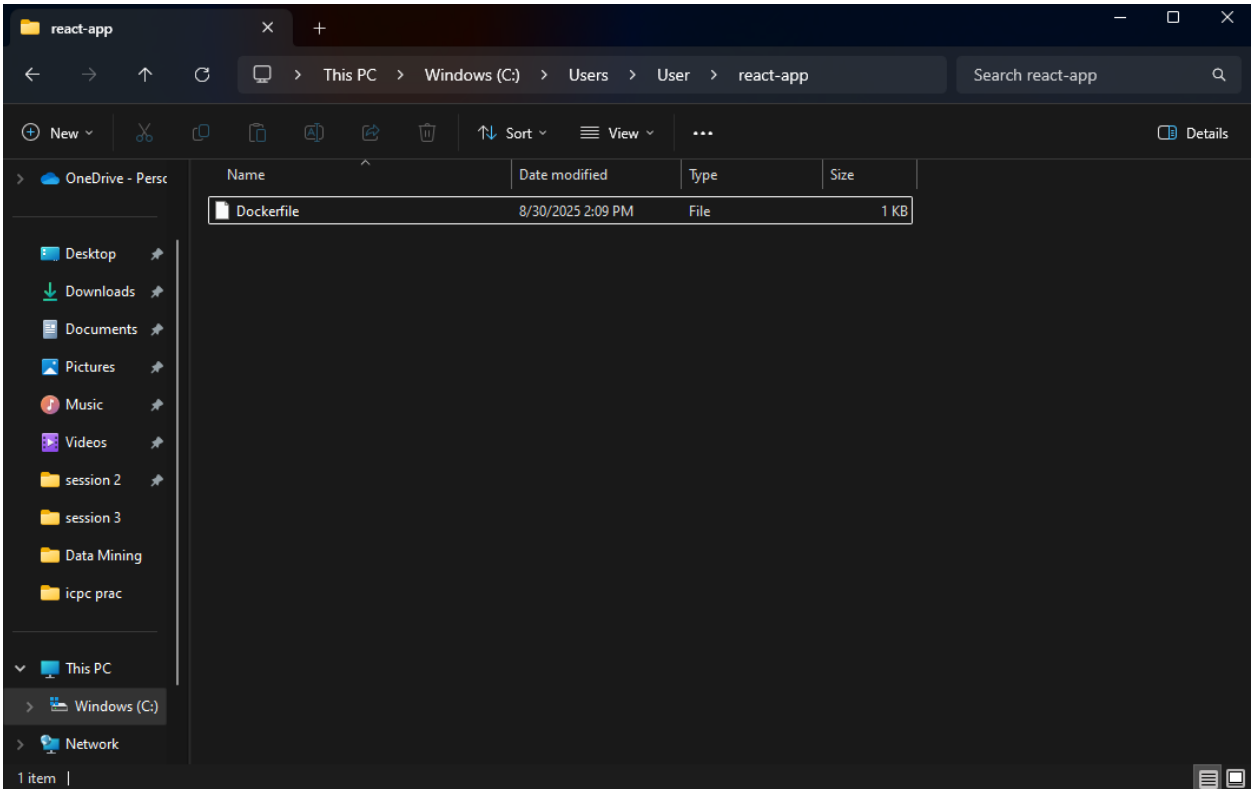
Verify variable



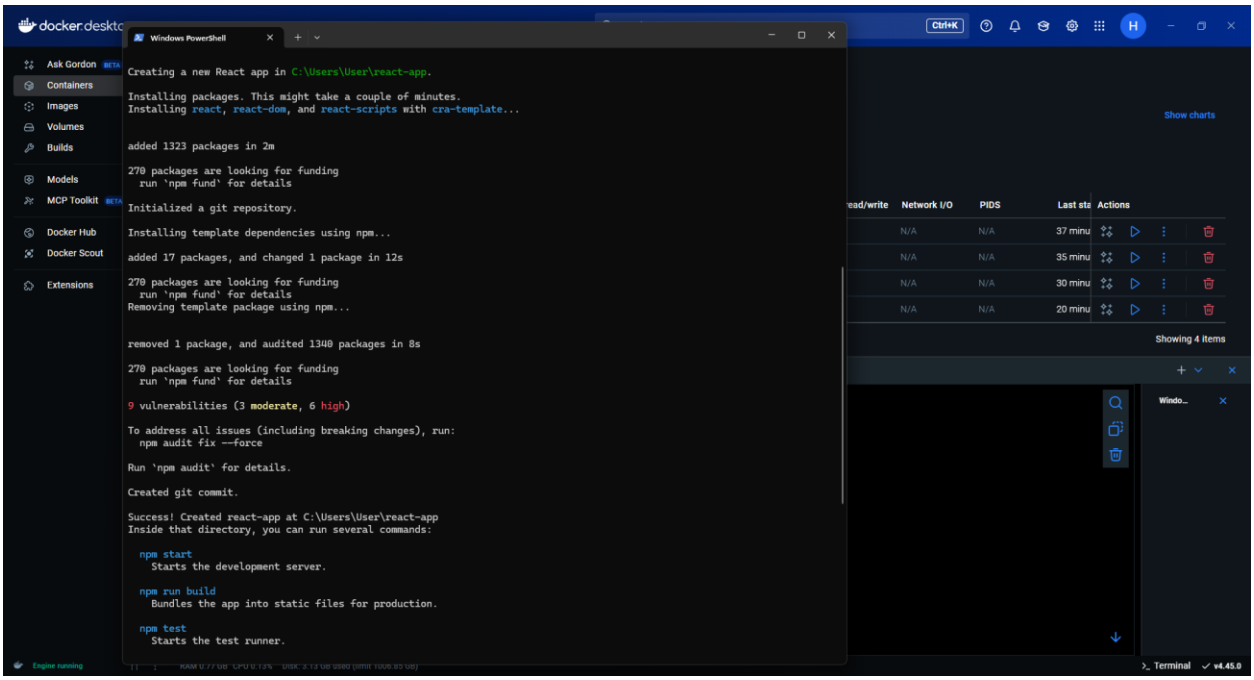
4. Docker Images

Create a Dockerfile in the root of your app repo (React app).

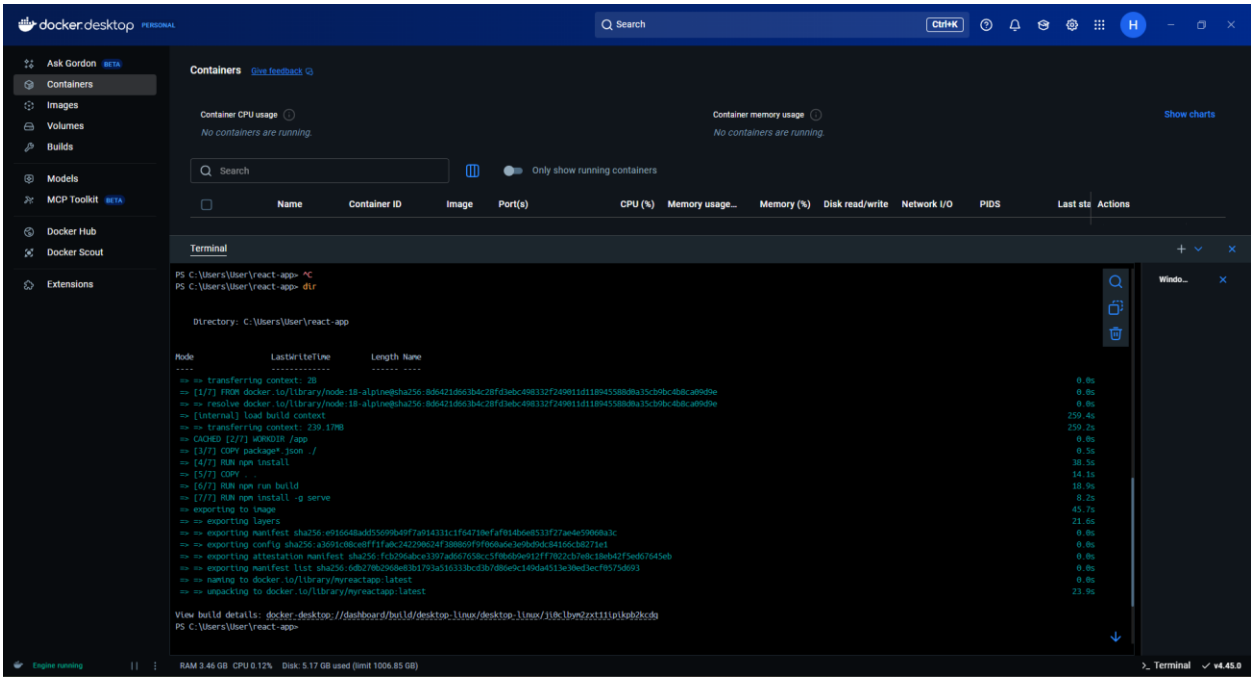




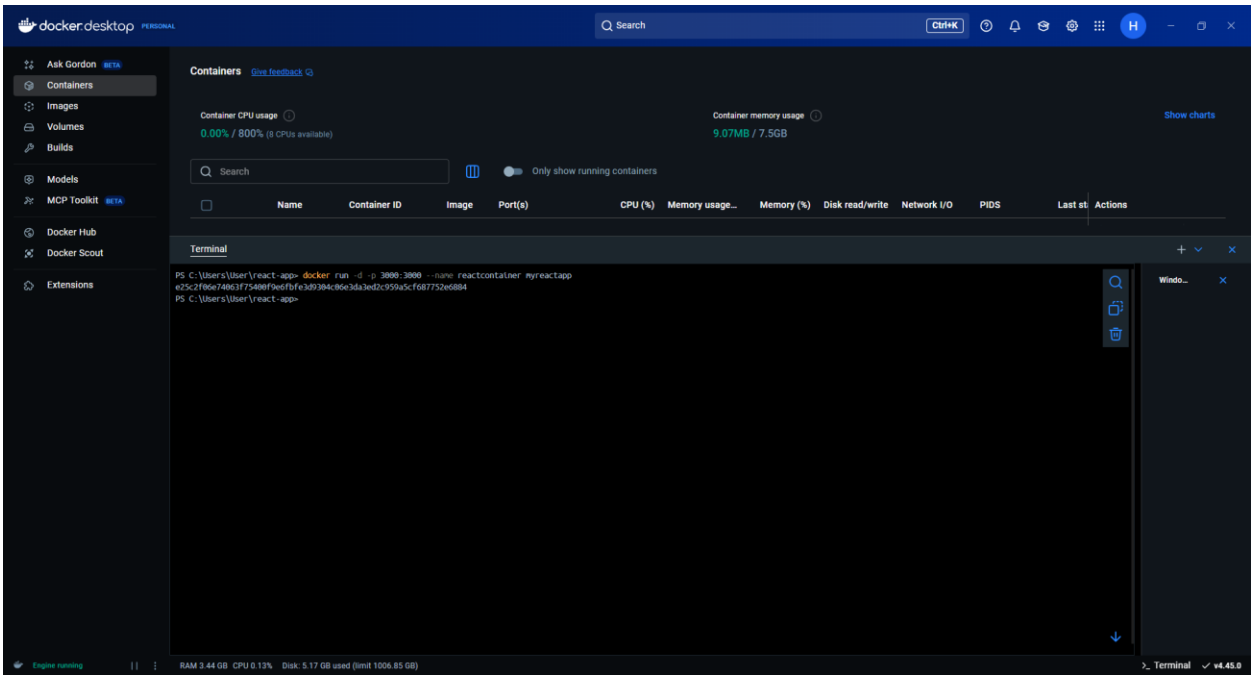
Create react app



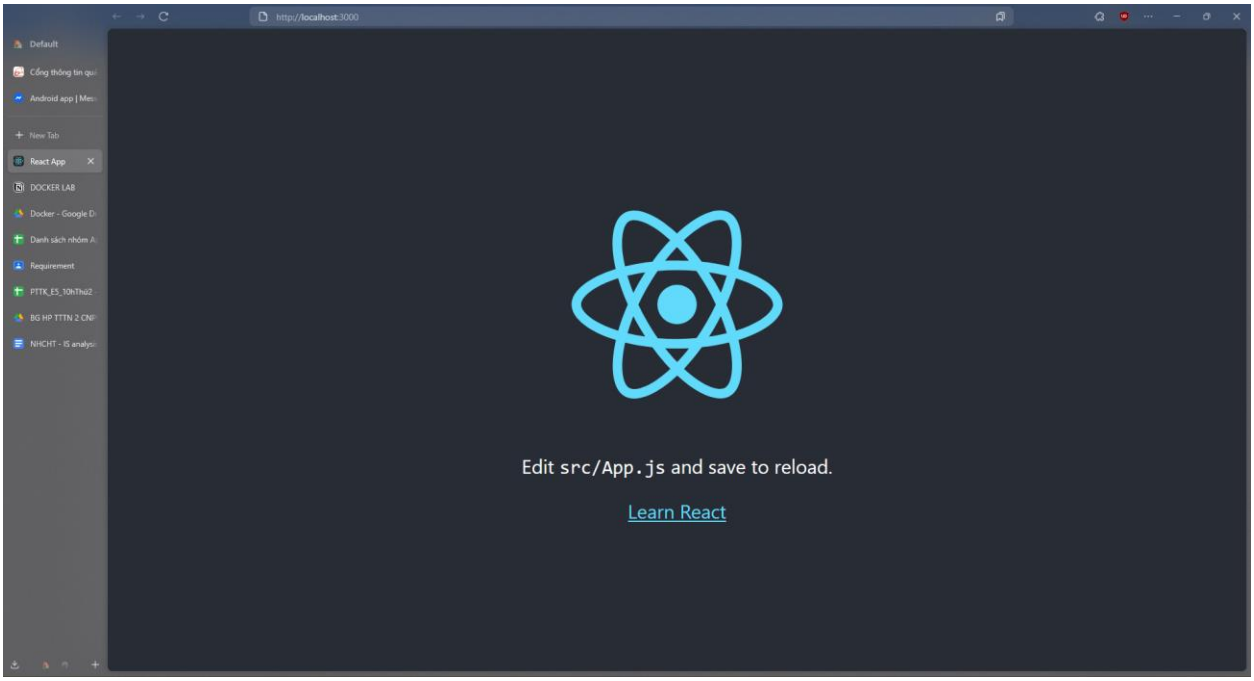
Build docker app with docker build -t myreactapp .



Run container

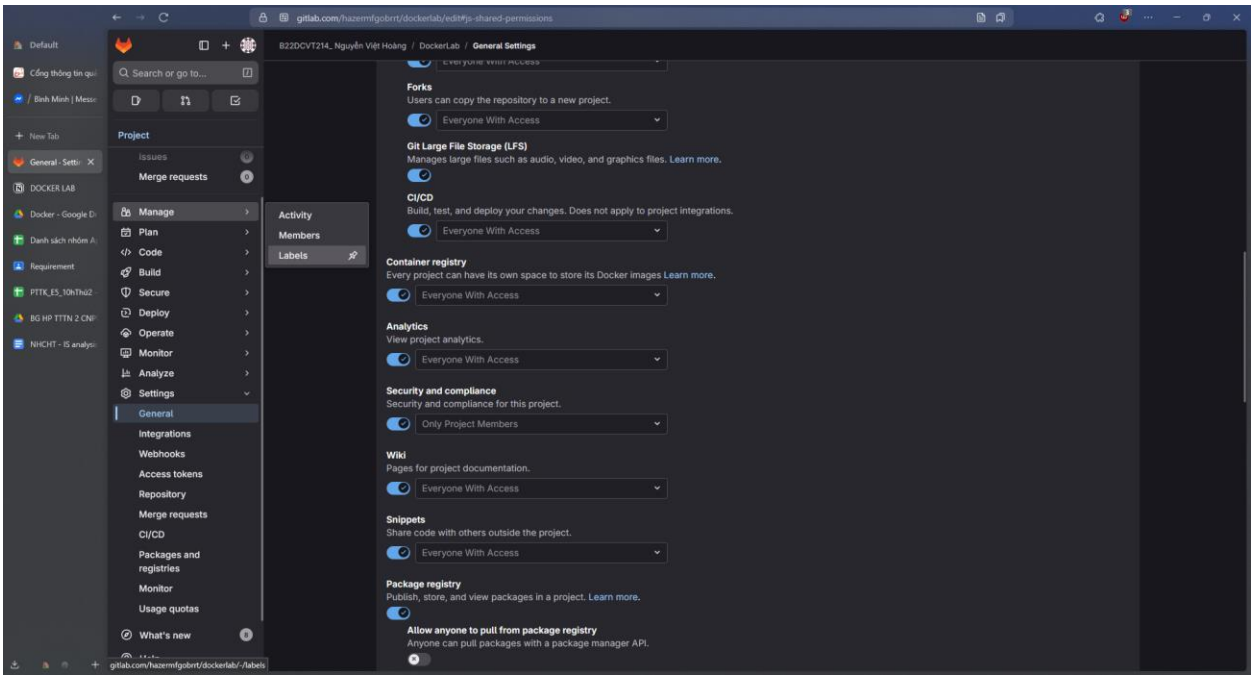


Test on browser

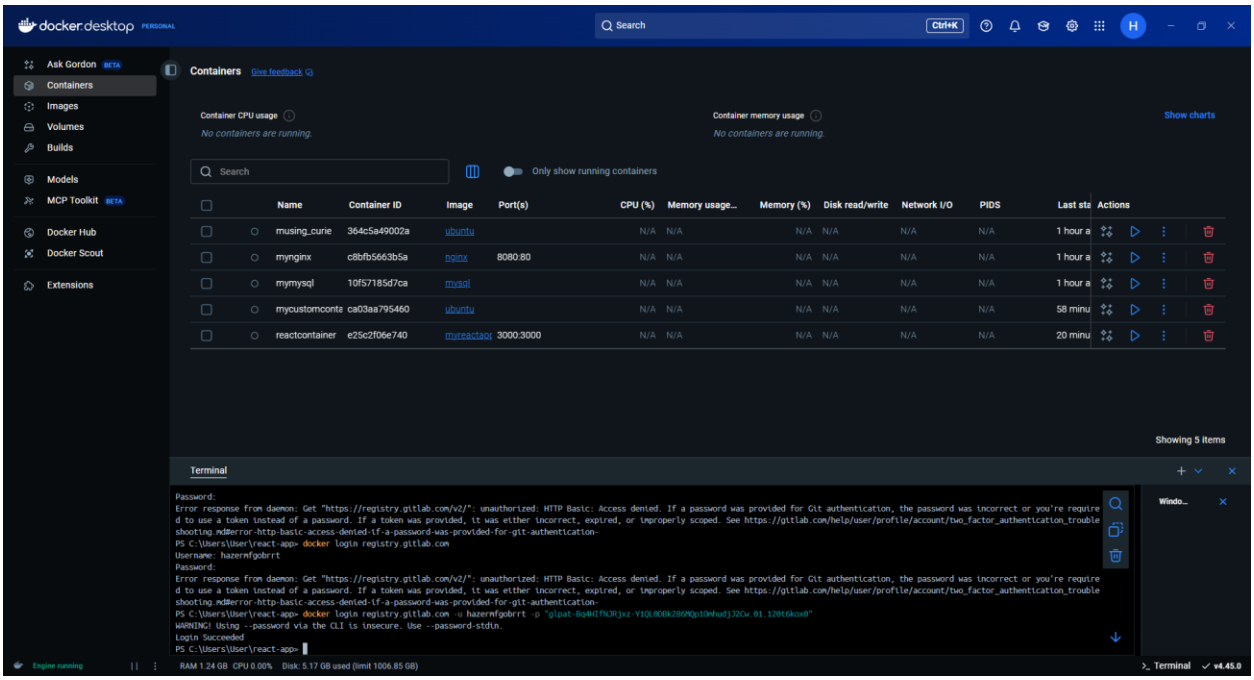


5. Docker Registry (GitLab Container Registry)

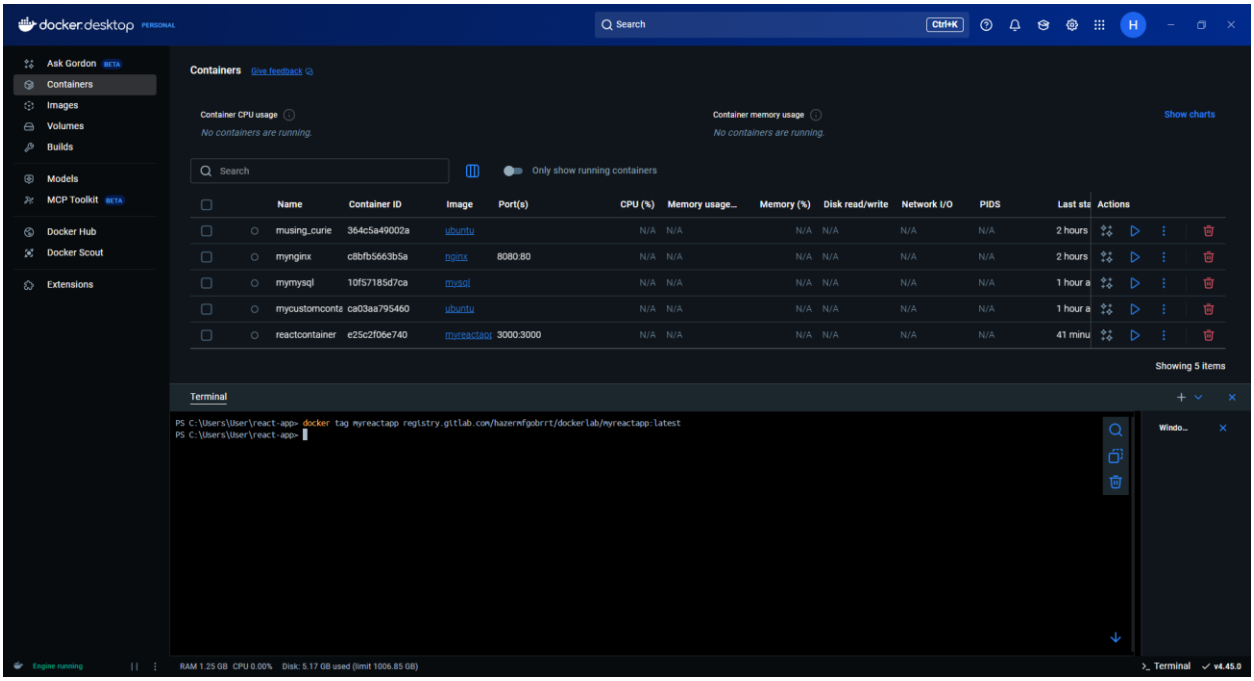
Enable GitLab Container Registry for your project.



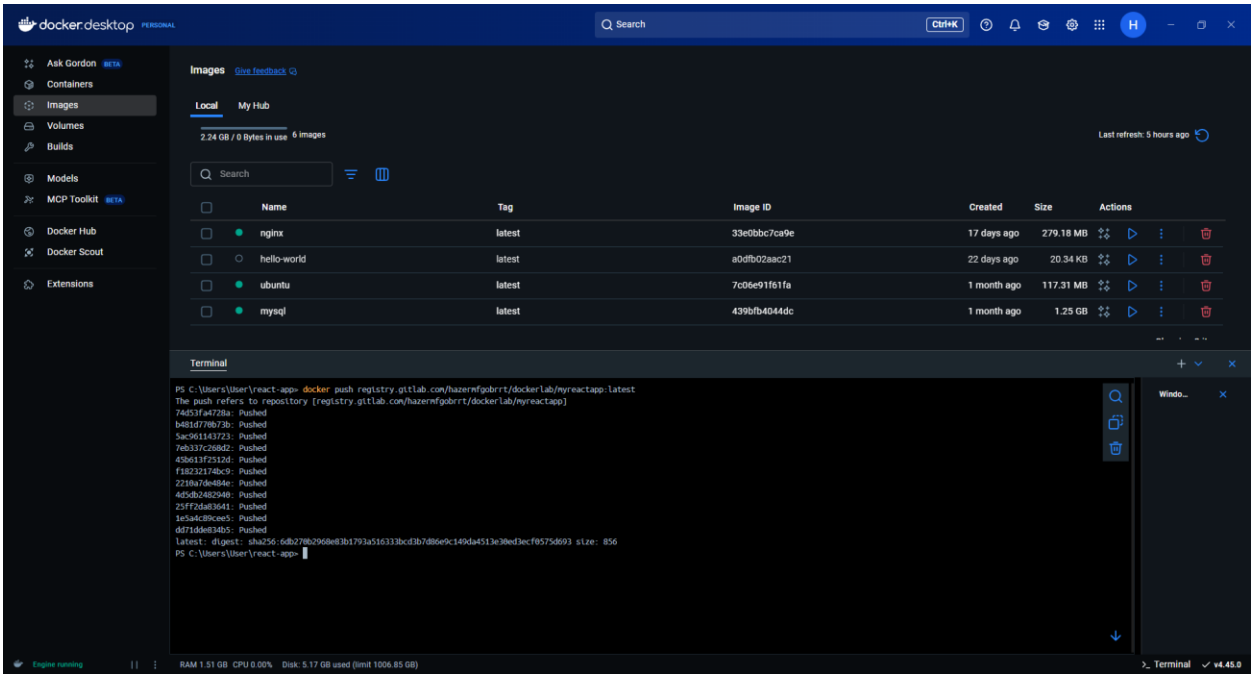
Login GitLab



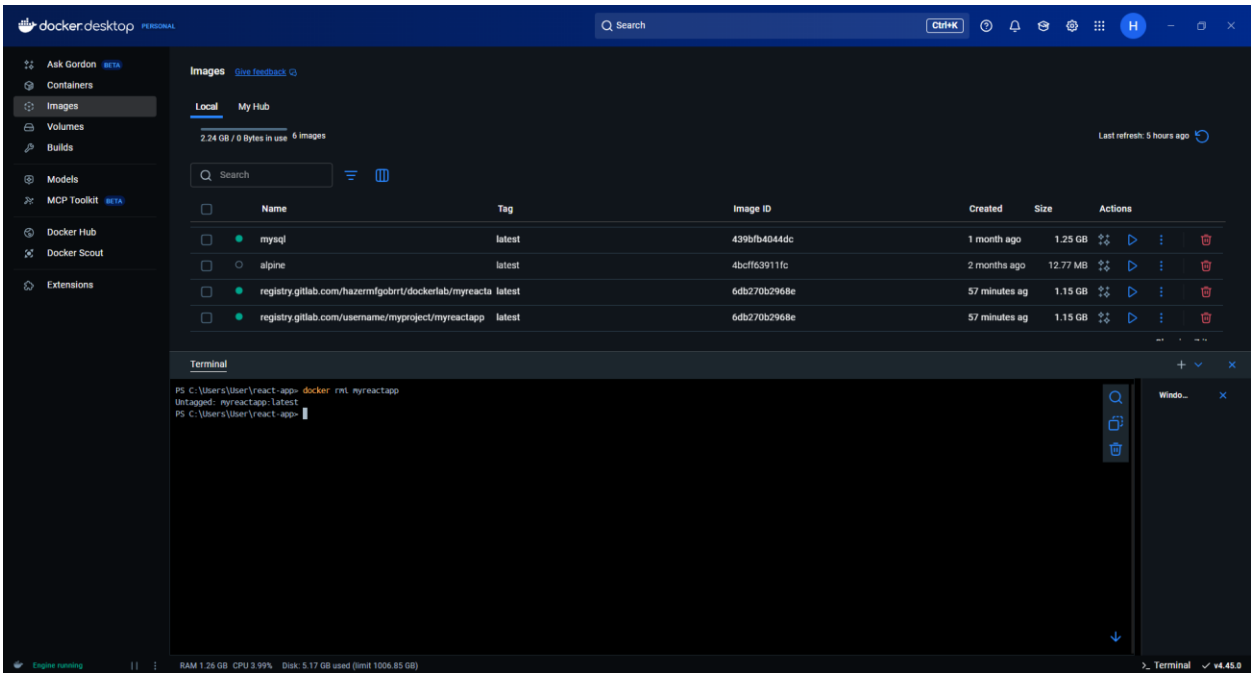
Tag local image



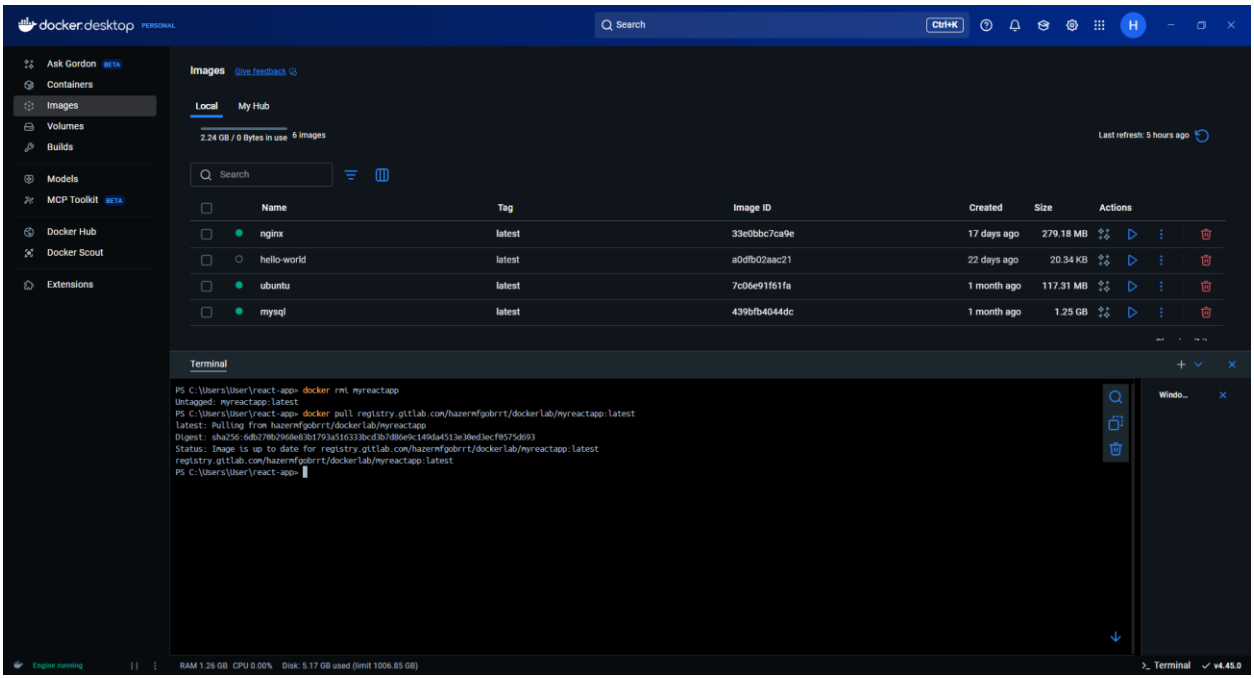
Push the image



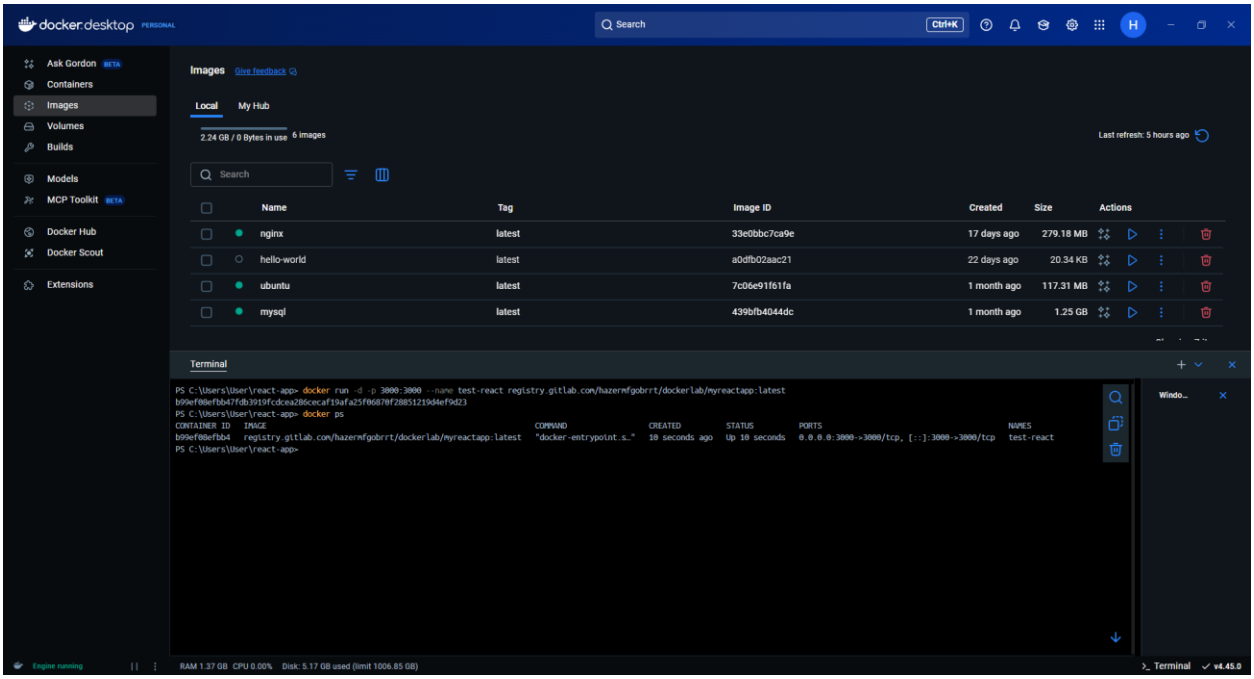
Remove local image



Pull the image back

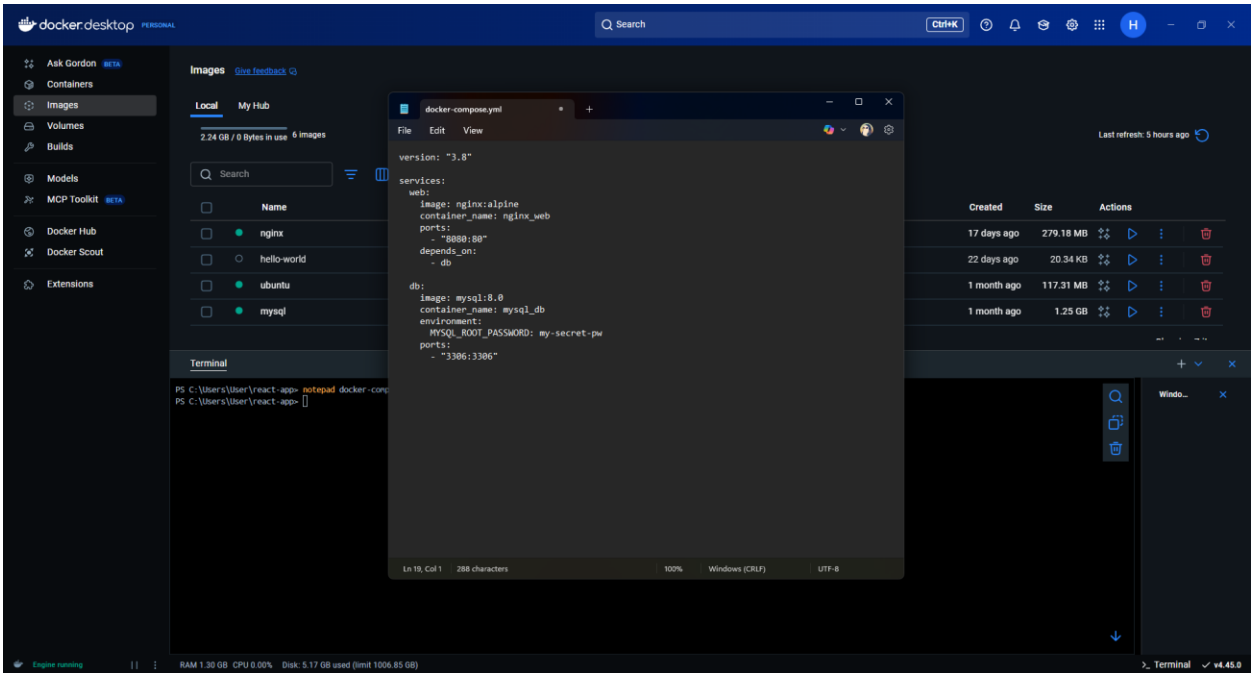


Run container

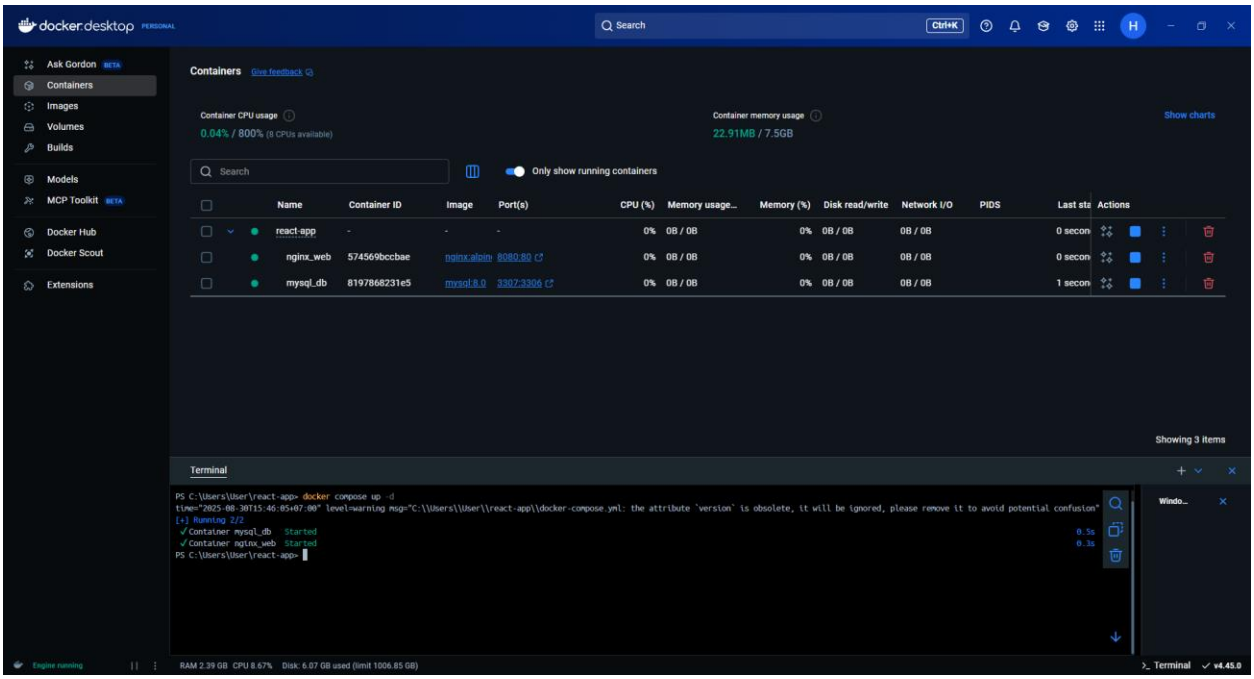


6. Docker Compose (Part 1 – Basics)

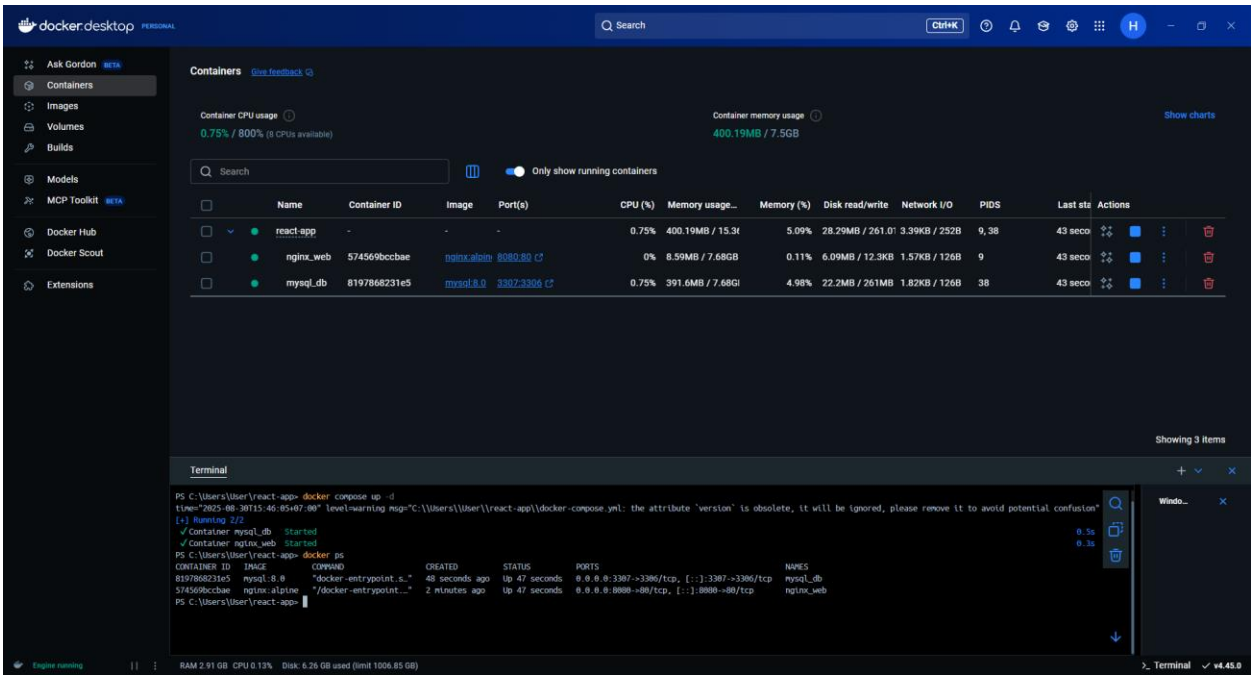
Create docker-compose.yml with services



Run docker compose up

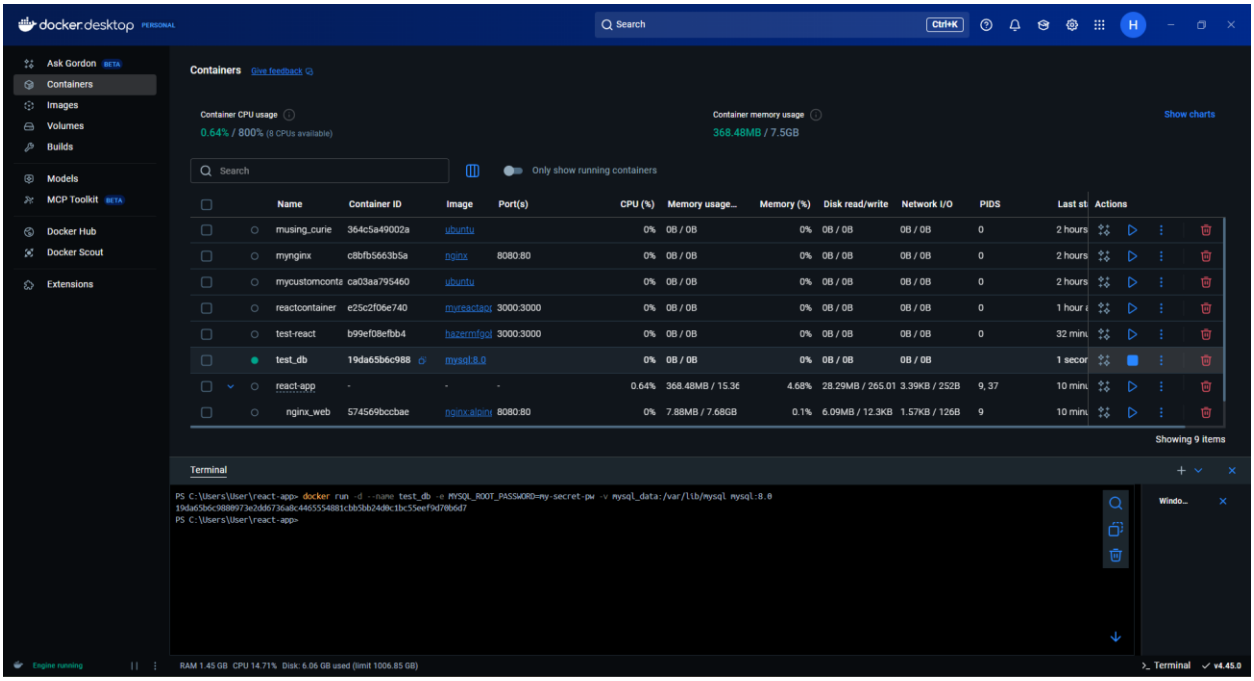


Verify both containers are running with docker ps

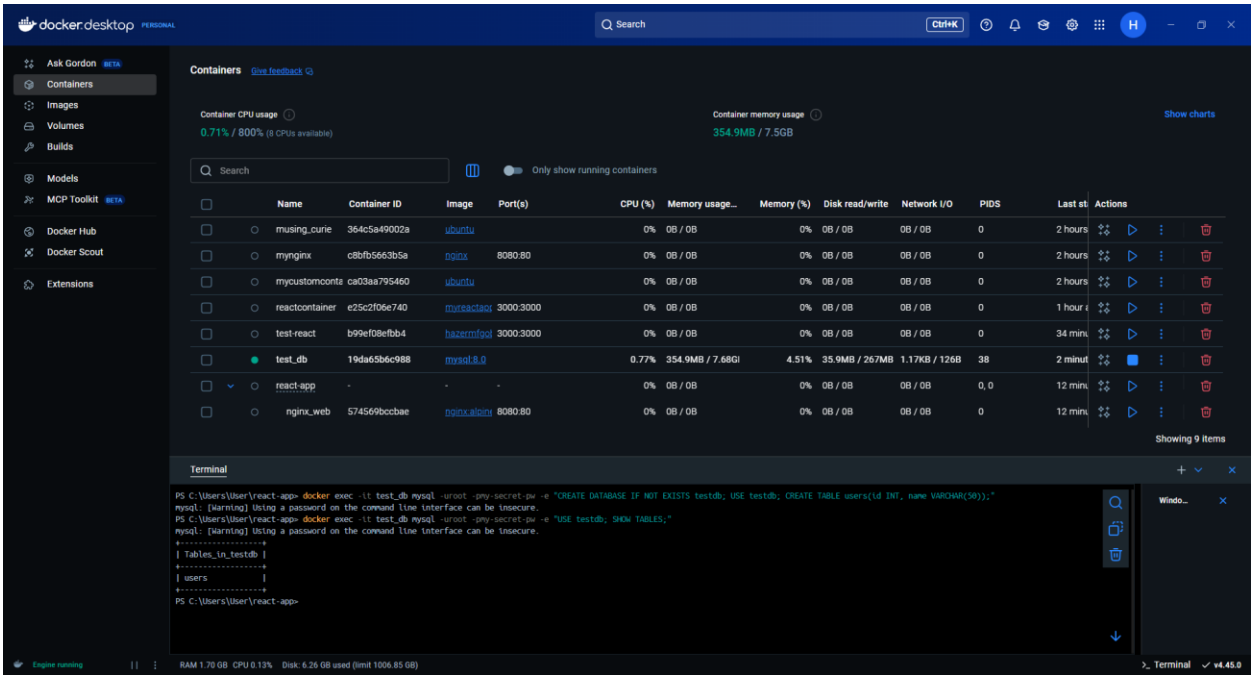


7. Docker Storage

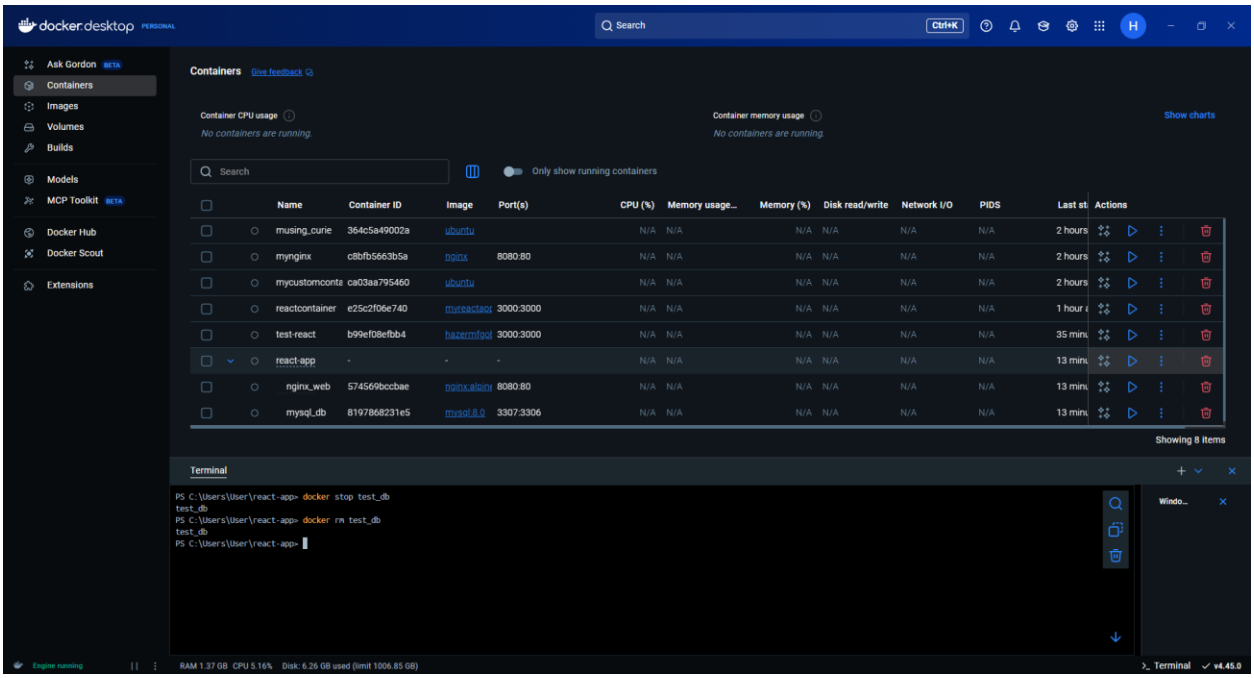
Run MySQL with a named volume



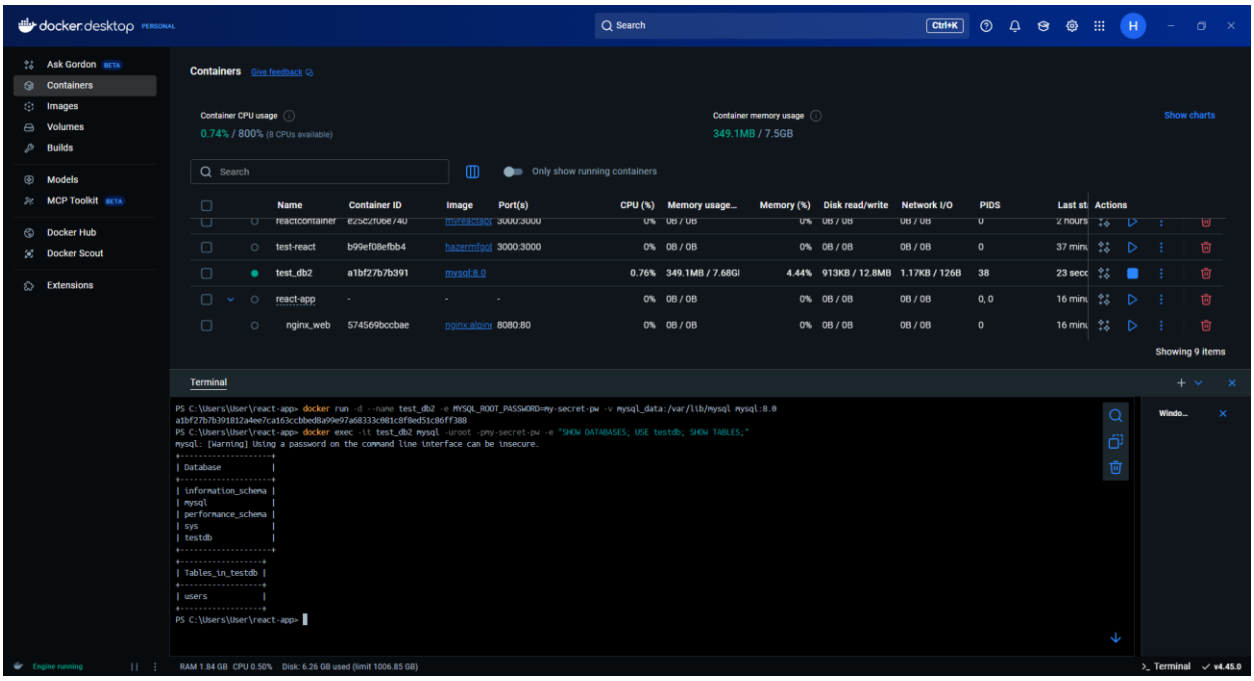
Create a table



Stop and remove the container

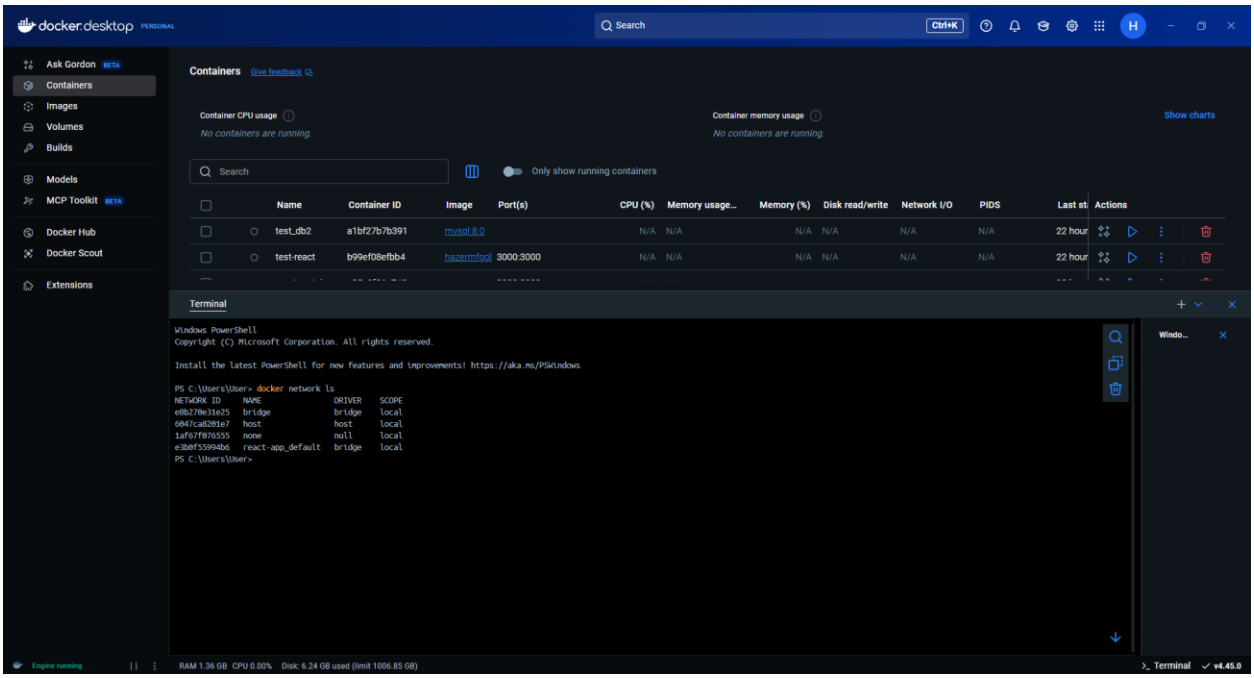


Start a new MySQL container using the same volume

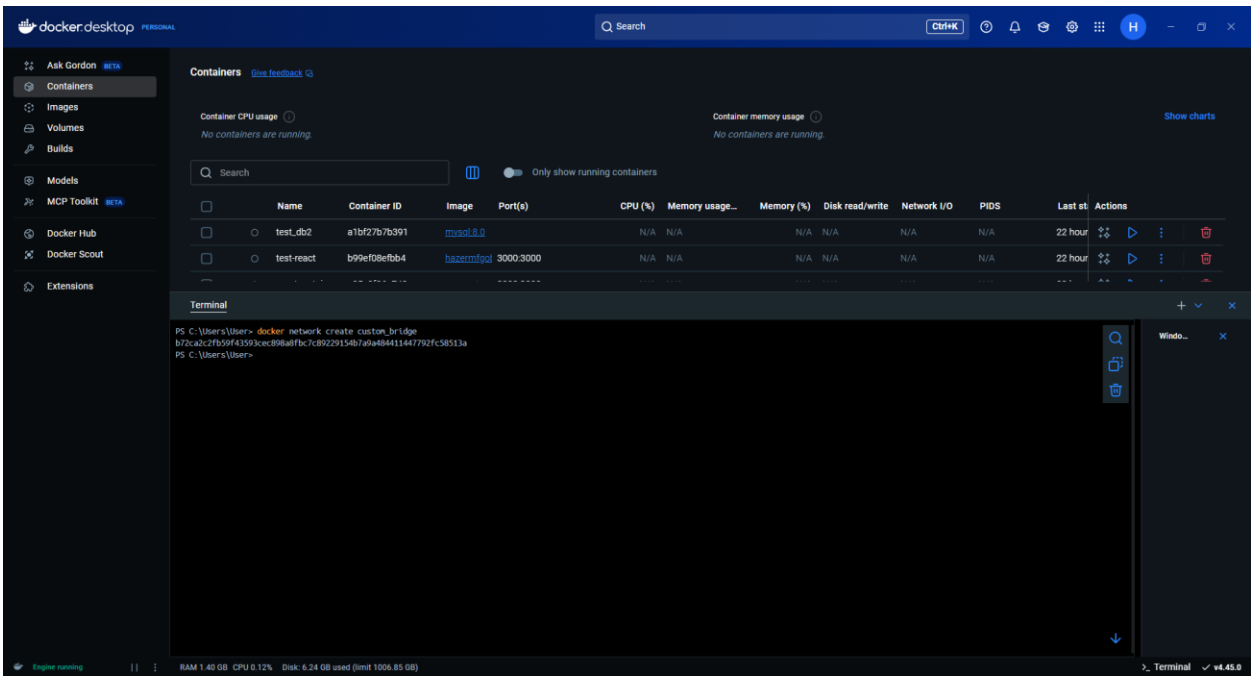


8. Docker Network

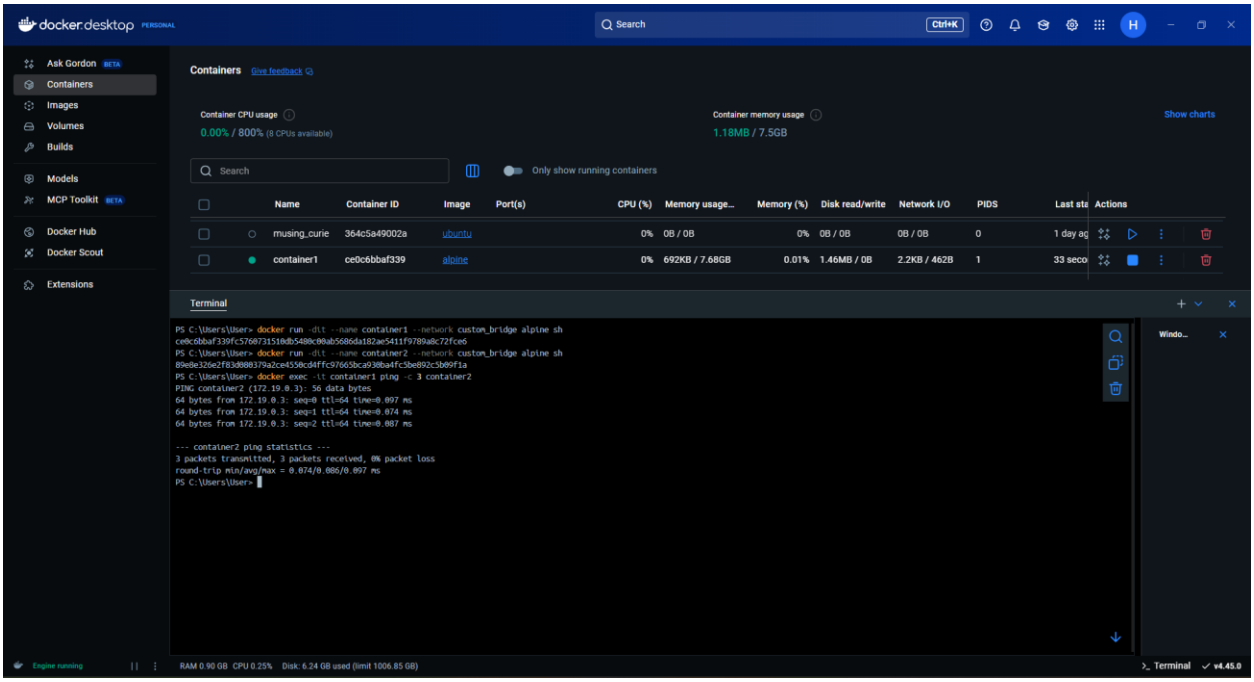
List networks with docker network ls



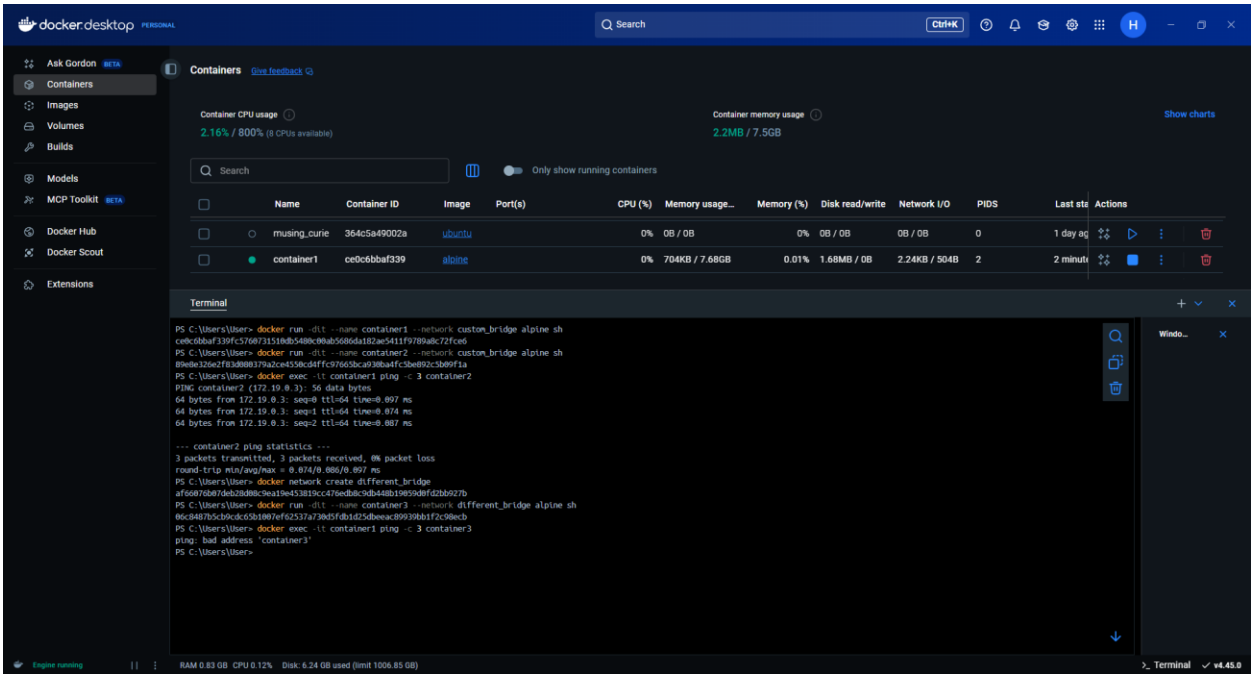
Create a custom bridge network



Run two containers in the same network and ping each other

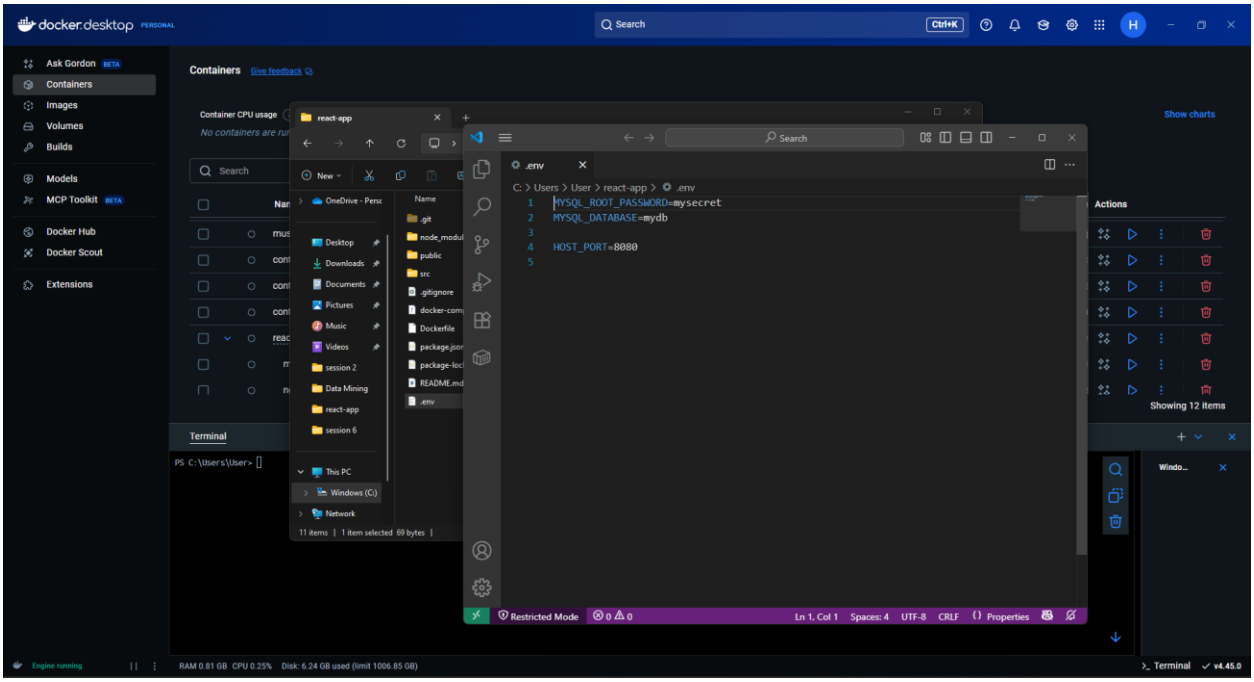


Run containers in different networks and test communication

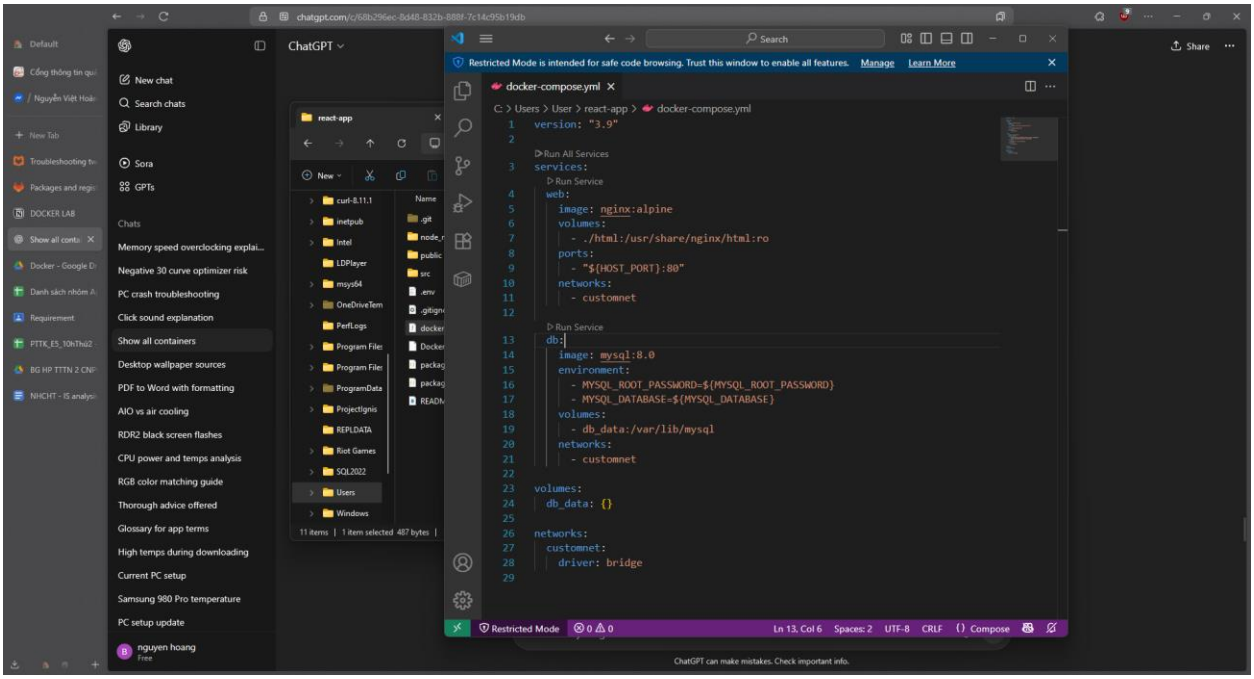


9. Docker Compose (Part 2 – Advanced)

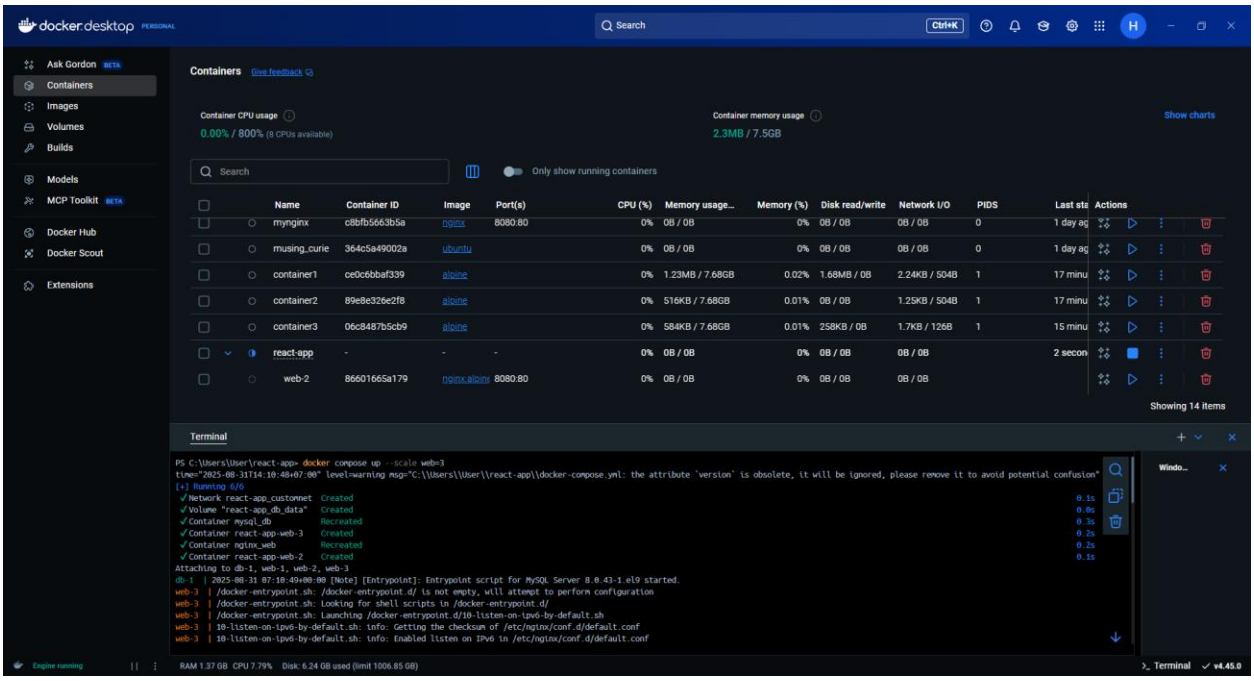
Add environment variables using a .env file



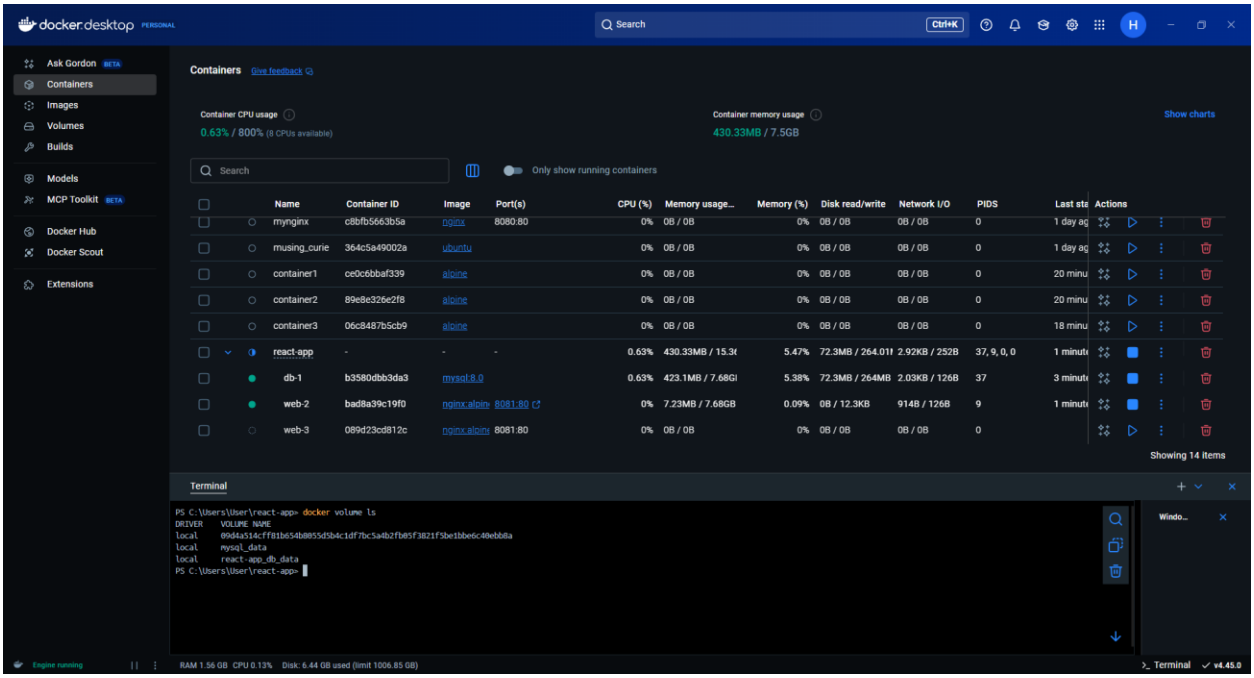
Add volume mappings for persistence and define custom networks in docker-compose.yml



Scale services with docker compose up --scale web=3

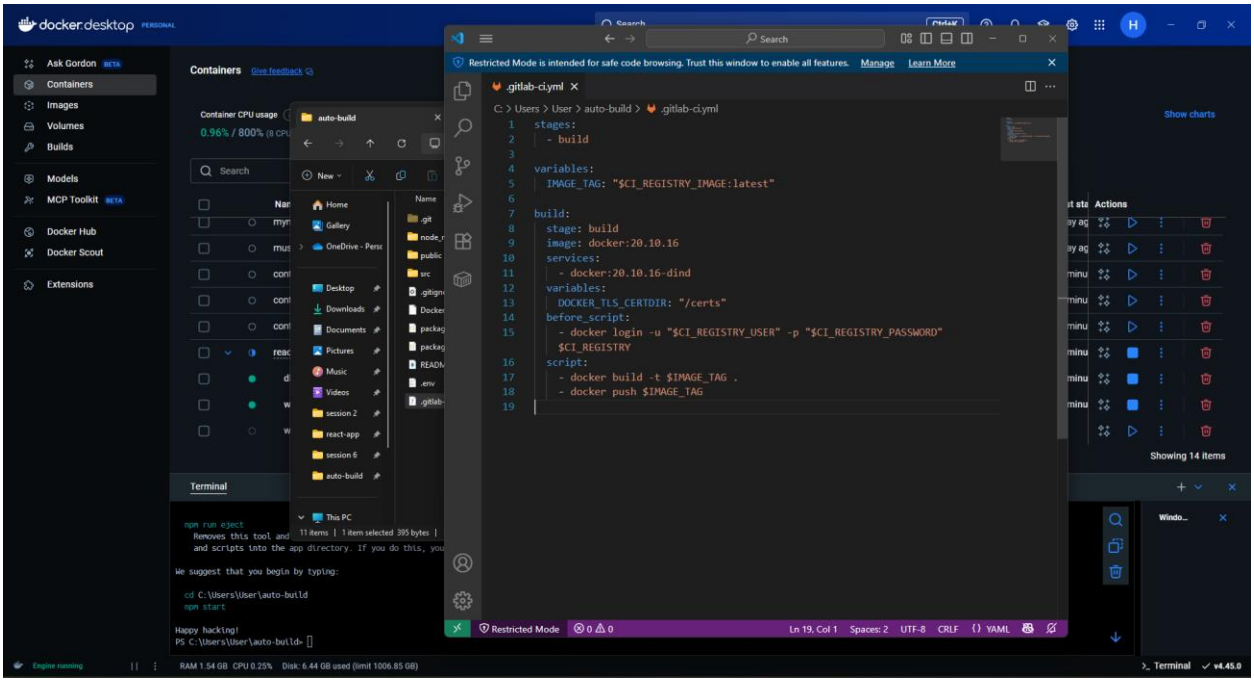


Check all the volume

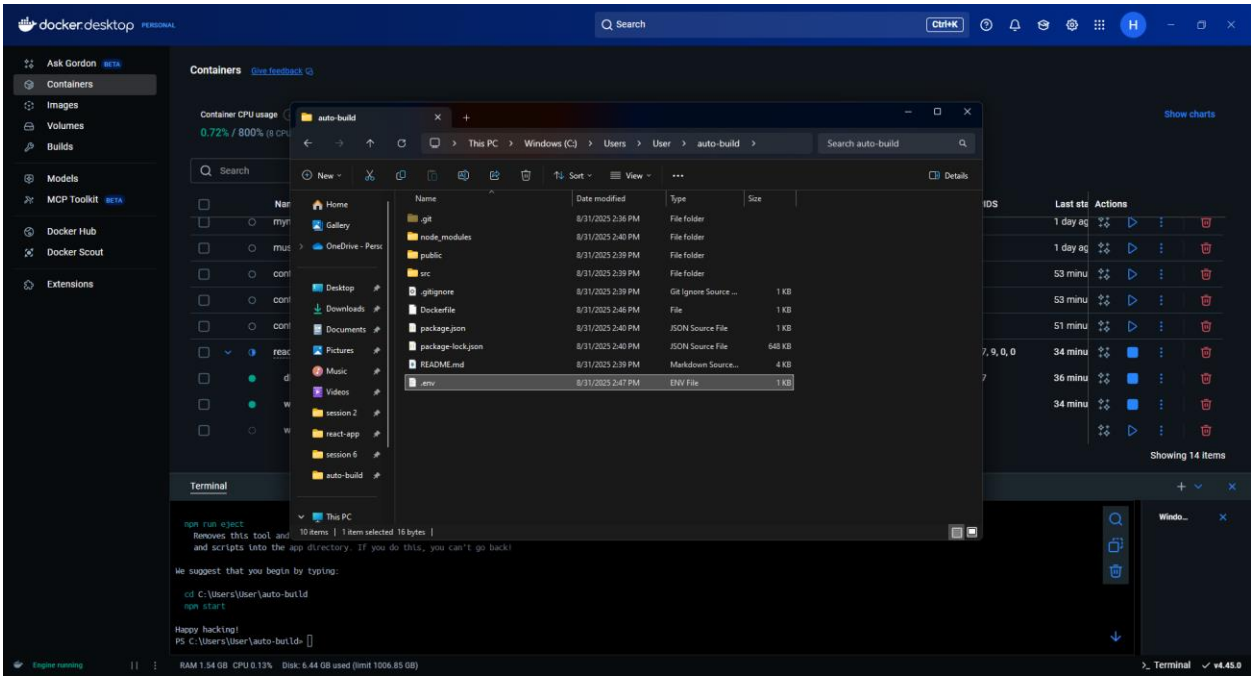


10. Advanced – Build & Push with GitLab CI/CD

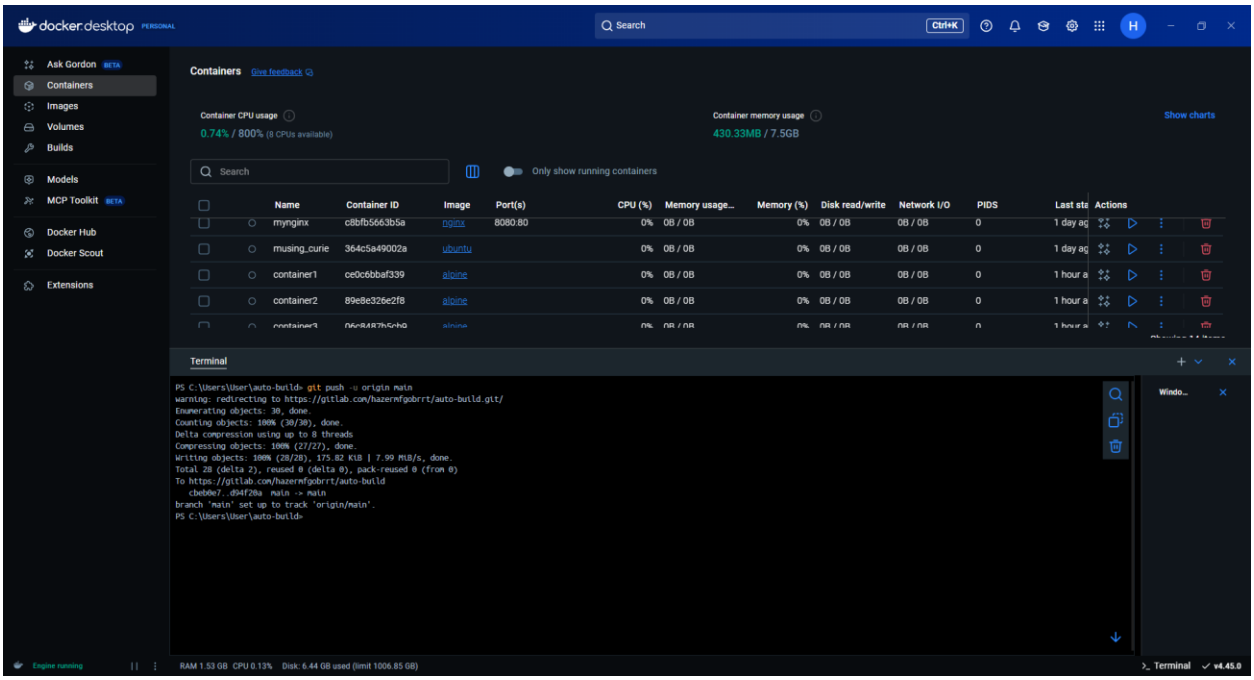
Create new GitLab repo



Add a Dockerfile (from Step 4) to your repo.



Commit and push to GitLab



Check GitLab CI/CD pipeline