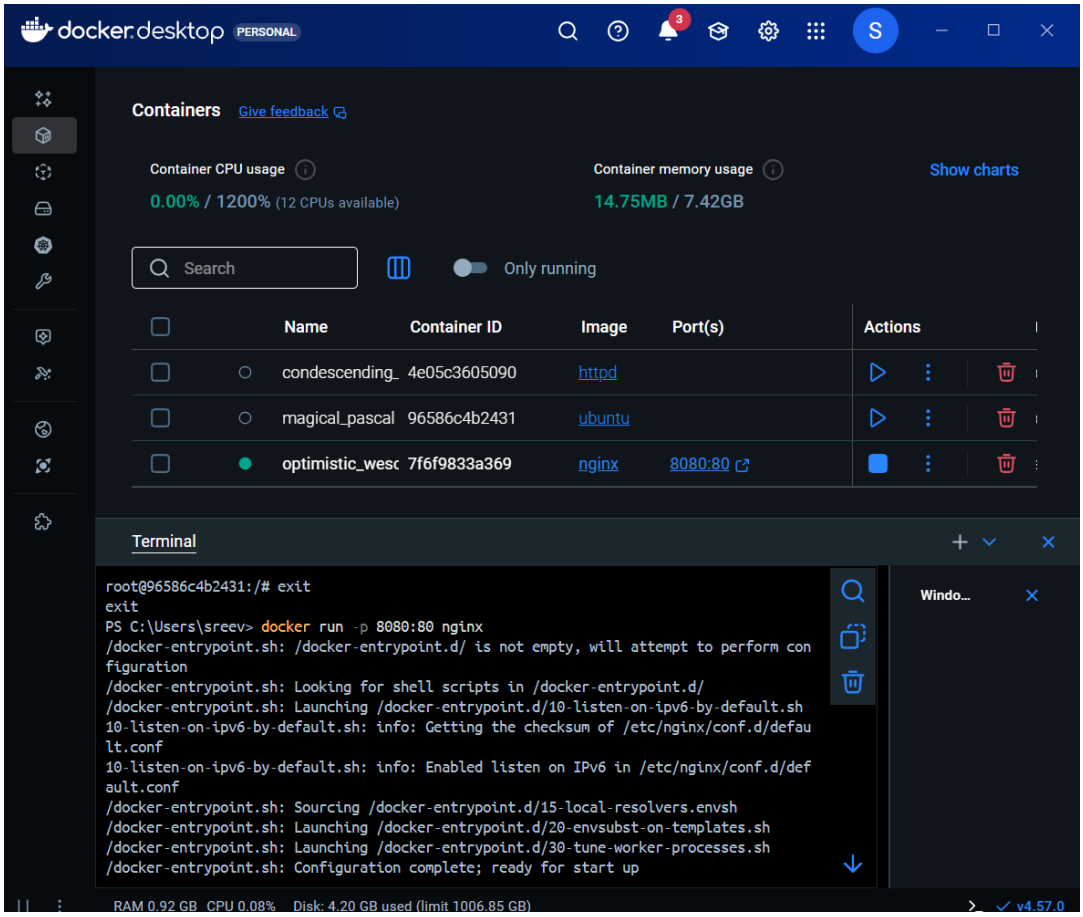
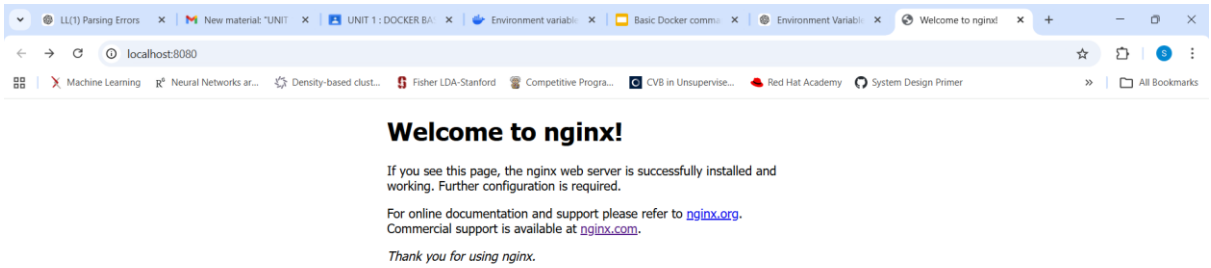
Learnings for lifetime:

1)Port publishing activity

So basically, covered publishing port mechanism, I must say it's just intricately beautiful though it might be very basic it really showed me how powerful docker is even at basic levels, so according to what I understood of it, since containers run in isolated namespaces the ports they use are inaccessible to users outside of it so we publish the port sort of exposing it to outside world by **publishing a port** which is Docker's way of intentionally breaking that isolation in a controlled manner: Docker makes the **host machine listen on a chosen port** and then **forwards all traffic arriving on that host port to the container's internal port**.

In other words, the container still thinks it is listening privately, but Docker transparently acts as a bridge between the external world and the container, exposing only what we explicitly allow.



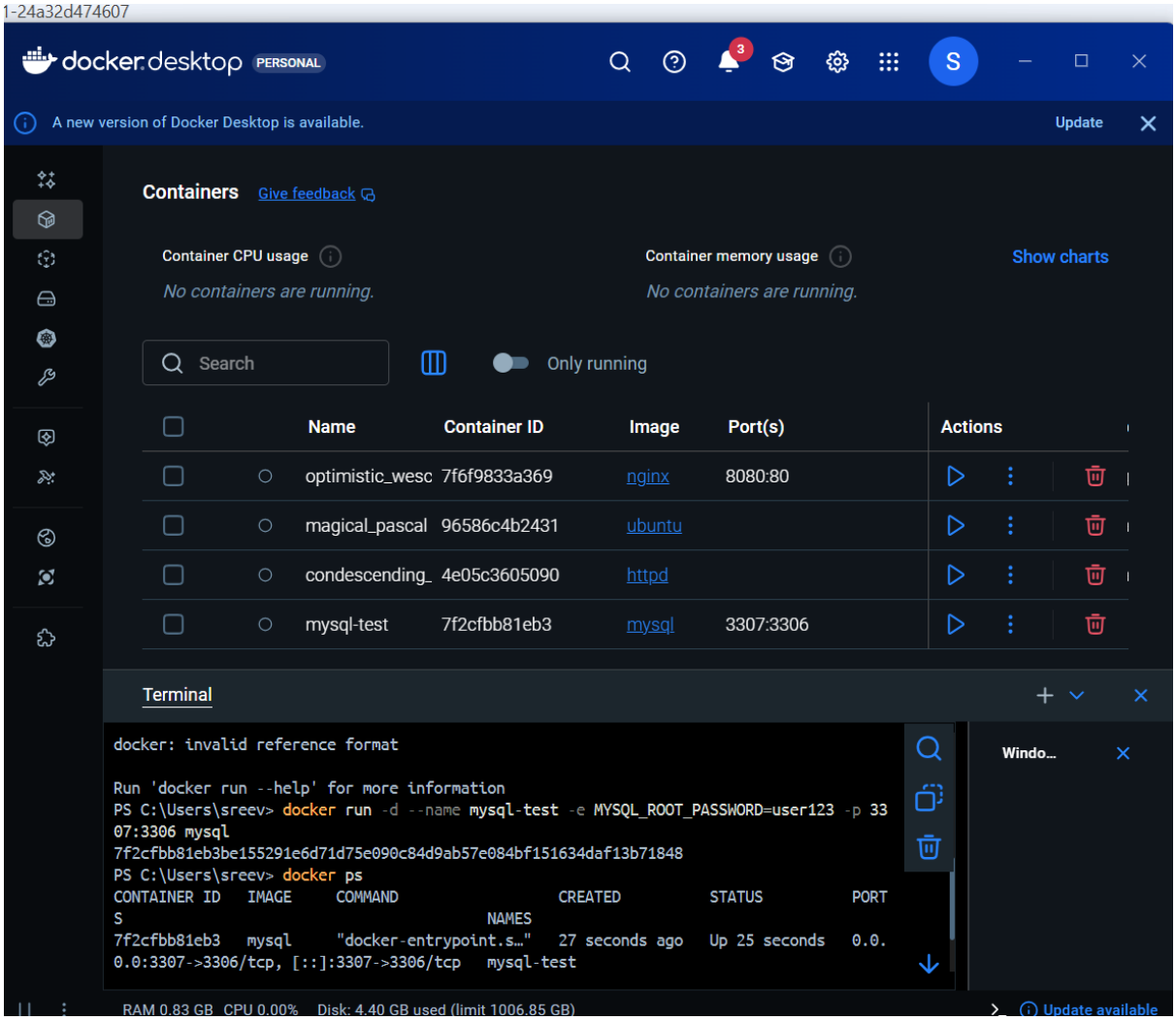


What happens internally?

Browser → localhost:8080 → Docker Host → Container:80 → Nginx

2)Some more interesting stuff ahead: started to work more on port publishing and MySQL container initialization.

Started exploring Docker port publishing and MySQL container initialization in more depth. By running a MySQL container in detached mode and setting the MYSQL_ROOT_PASSWORD environment variable, I understood how Docker images use environment variables during initialization to configure services automatically. I also learned that publishing a port (-p 3307:3306) does not affect how services work *inside* the container, but instead allows the host machine to access the MySQL server running in the container. Using docker exec, I was able to run the MySQL client inside the container without relying on port publishing, which clarified the difference between internal container access and external host access. Overall, this showed how Docker maintains isolation by default while still allowing controlled exposure of services when needed.




```
PS C:\Users\sreev> docker exec -it mysql-test mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.6.0 MySQL Community Server - GPL

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
PS C:\Users\sreev>
```

When I publish a port in Docker, I'm not "opening a port inside the container." I'm asking Docker to stand at the host's door, listen on my behalf, and quietly forward anything it hears into the container's private world.
Behind the scenes, Docker sets up all the low-level networking machinery—NAT rules, packet forwarding, and bridge connections—so the container

remains isolated, yet reachable, without me ever touching iptables or firewall configurations.