🚢 **Docker Learning – Day 2: Images, Containers & Sharing to the World** Today was all about understanding how Docker really works under the hood — and using it like a real developer. Here’s what I covered:

### **🧱 What Are Docker Images?**

I learned that a **Docker image** is like a recipe for your application. It contains everything needed to run the app — code, dependencies, and environment.  
 Images are created using a **Dockerfile**, and each instruction in that file adds a new “layer” to the image. Once built, the image is **immutable**, meaning it won’t change unless rebuilt.

### **📦 Containers – The Running Version**

A **container** is the running instance of an image. You can start, stop, restart, and even run multiple containers from the same image.  
 There are two ways to run containers:

* **Attached mode**, where you see the output in your terminal.
* **Detached mode**, where it runs in the background.

You can also bind ports, give custom names, or pass environment variables when running a container.

### **🔧 Managing Containers & Images**

I practiced listing, starting, stopping, and removing both **images and containers**.  
 Also explored useful commands to inspect what’s going on inside a container, and how to clean up unused resources.

### **🔍 Debugging with Logs & Shell Access**

When things don’t go right, I now know how to **check logs** of a container or **get inside it using a terminal**. This helps a lot while debugging or inspecting container behavior live.

### **☁️ Docker Hub – Sharing My App**

The most exciting part was learning how to **tag and push** my custom image to **Docker Hub** — a public place to store Docker images.  
 Now anyone can pull and run my app with just one command — without needing to set up the project themselves.

### **🧠 Key Takeaways from Day 2:**

* **Images** are blueprints; **containers** are live apps.
* You can manage everything with a few simple Docker commands.
* Sharing apps via Docker Hub makes deployment super fast and universal.
* You don’t need to install Node.js or dependencies if it's already inside the image — Docker handles everything.

Tomorrow, I’ll jump into **Docker Volumes** and **Docker Compose** — time to explore data persistence and multi-container apps.  
 If you're learning Docker too, feel free to connect and share your journey!

#Docker #DevOps #LearningInPublic #Containers #CloudComputing #DockerHub