5.06: Exploring Docker Essentials and Container Lifecycle

LAB *Obligatorio

Lab Overview

In this lab, you will apply the **fundamental Docker concepts** learned in the first three lessons of Day 3. Over approximately two hours, you'll **verify** your Docker installation, **run multiple containers** simultaneously, **inspect** their states, and **practice** stopping/removing them cleanly. You'll also explore **container isolation** and see how each container can operate independently on the same system.

This is a **hands-on** lab that encourages you to figure out the exact commands and parameters rather than following explicit step-by-step examples. The aim is to **reinforce** your understanding of Docker's architecture and CLI commands while keeping you engaged throughout the process.

Lab Objectives

By completing this lab, you will be able to:

- **Confirm** Docker is set up and functioning on your machine.
- **Run** multiple containers at once (e.g., a web server and a database) without conflicts.
- **Investigate** container logs, networking, and resource usage to understand container isolation.
- **Practice** proper cleanup by stopping and removing containers and images.
- **Document** your findings and reflections in a Google Doc.

Instructions

EN

> referencing the exact Docker commands from your lesson notes. Instead, recall the core commands and flags you've learned (e.g., run, ps, logs,

> stop, rm, rmi, etc.) and find the correct usage. Feel free to consult the official Docker documentation if needed.

Task 1: Verify Docker Installation

1. **Check Docker Version**
   - Confirm Docker is recognized in your terminal (PowerShell, Command Prompt, or macOS Terminal).
   - You should see a short version output.
2. **Run a Quick Container**
   - Pull and run a minimal “hello-world” style image.
   - Observe the output to ensure Docker can pull and run images successfully.
3. **Discuss**
   - In your notes, briefly record how you confirmed Docker was installed (e.g., version check, output screenshot).

Task 2: Pull and Run Multiple Containers

1. **Select Two Images**
   - Choose any two official images from Docker Hub (e.g., one web server, one database, or any combination that interests you).
   - Recall how to fetch them.
2. **Run Them**
   - Launch both containers in a way that they stay up in the background (detached).
   - Ensure you map ports correctly if your containers need external access (for instance, 8080 for a web container).
3. **Check**
   - Confirm both containers are running simultaneously without port conflicts or errors.

> **Hint**: If you pick a web server container, try accessing its default page via `:<mapped_port>` in your browser.

Task 3: Investigate Containers

1. **Inspect**
    - View container details (e.g., environment variables, IP addresses, volumes).
    - Notice how Docker sets up networking, assigns container IDs, etc.
2. **View Logs**
    - Check each container's logs.
    - If your containers produce minimal logs, consider adding flags or environment variables that generate extra output to see something meaningful in the logs.
3. **Optional**: Explore Inside a Container
    - If one container has a shell (e.g., `bash`), attach to it.
    - Observe differences between the container environment and your host system.

> **Hint**: For more insight, see how the container's process list differs from your host's.

Task 4: Container Lifecycle Management

1. **Stop**
    - Gracefully stop both containers.
    - Observe how Docker waits for the container's main process to end (or kills it if it's unresponsive).
2. **Remove**
    - Remove both containers now that they're stopped.
    - Verify they no longer appear in your container list.
3. **Clean Up Images**
    - If you won't reuse the images, remove them to free disk space.
    - m your local image list is updated accordingly.

> **Hint**: Use filtering or referencing container IDs if you have many containers or images.

Task 5: Document Your Results

1. **Screenshots**
    - Capture critical steps (running multiple containers, logs, final cleanup).
    - Show them in your **Google Doc** with brief captions or explanations.
2. **Short Reflection**
    - Write 1–2 paragraphs summarizing any **challenges**, **surprises**, or **key takeaways**.
    - Example prompts:
        - "Did anything unexpected happen when running containers simultaneously?"
        - "How did container isolation differ from your host environment?"
3. **Share the Link**
    - Ensure your Google Doc is publicly accessible (or shareable with your instructor).
    - Submit the link in the provided submission form.

Conclusion

By finishing this **Docker Essentials & Container Lifecycle** lab, you have:

- **Verified** your Docker environment is functional.
- **Launched multiple containers** simultaneously and inspected them for logs and process details.
- **Practiced** stopping/removing containers and tidying up images.
- **Documented** your experience in a Google Doc, sharing both screenshots and personal reflections.

This hands-on approach cements your **Day 3** learning. In future labs, you'll tackle **more advanced Docker usage**—like containerizing a Java application, setting up multi-container workflows with Docker Compose, and exploring Docker networking in greater depth.

Exploring Docker Essentials and Container Lifecycle

Submit the link to your solution below:

URL

Send lab

ANTERIOR
Installing Docker and Docker Commands

SIGUIENTE
Deploying a Web App as a Docker Container