

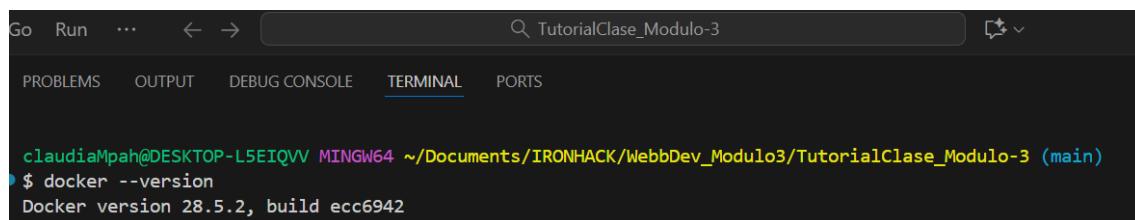
5.06: Exploring Docker Essentials and Container Lifecycle

LAB *Obligatorio

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.



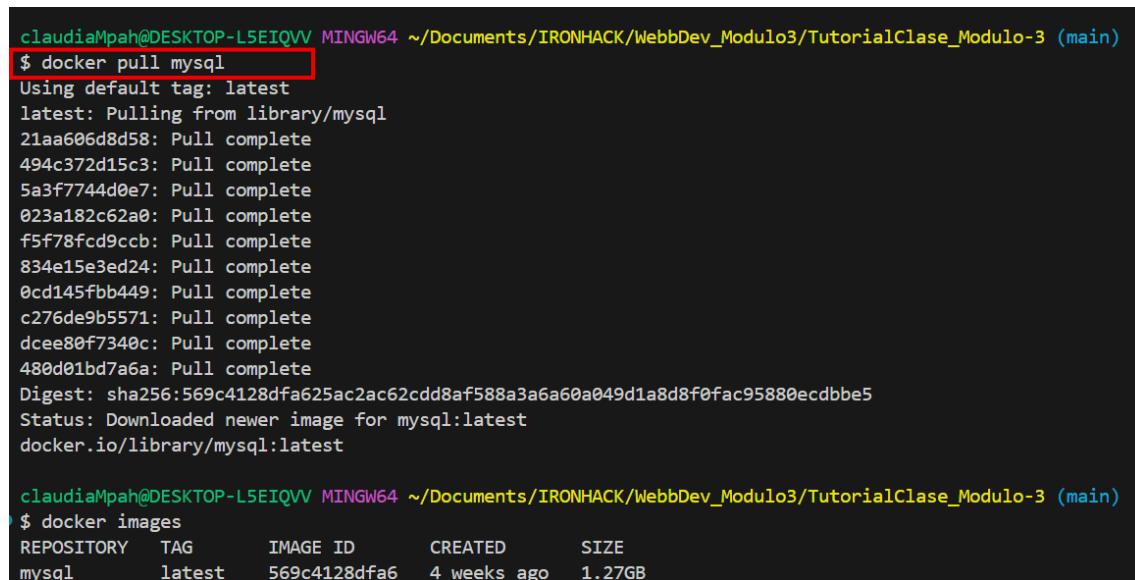
A screenshot of a terminal window titled "TutorialClase_Modulo-3". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal content shows a command-line session:

```
claudiaMpaah@DESKTOP-L5EIQVV MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker --version
Docker version 28.5.2, build ecc6942
```

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.

I decided to pull mysql image



A screenshot of a terminal window titled "TutorialClase_Modulo-3". The terminal content shows a command-line session:

```
claudiaMpaah@DESKTOP-L5EIQVV MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
21aa606d8d58: Pull complete
494c372d15c3: Pull complete
5a3f7744d0e7: Pull complete
023a182c62a0: Pull complete
f5f78fc9ccb: Pull complete
834e15e3ed24: Pull complete
0cd145fbb449: Pull complete
c276de9b5571: Pull complete
dcee80f7340c: Pull complete
480d01bd7a6a: Pull complete
Digest: sha256:569c4128dfa625ac2ac62cdd8af588a3a6a60a049d1a8d8f0fac95880ecdbbe5
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

claudiaMpaah@DESKTOP-L5EIQVV MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
mysql           latest        569c4128dfa6  4 weeks ago   1.27GB
```

Docker mysql image created:



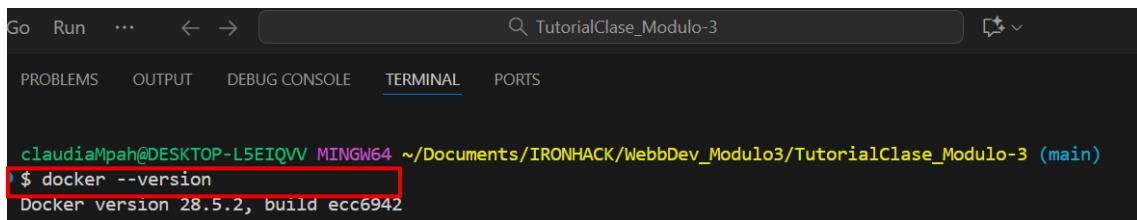
The screenshot shows the Docker Images interface. At the top, there are tabs for 'Images' (selected), 'Give feedback', 'Local', and 'My Hub'. Below the tabs, it says '996.67 MB / 1.27 GB in use 1 Images'. A search bar with placeholder 'Search' and a refresh button are on the right. The main area displays a table with one row:

	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	mysql	latest	569c4128dfa6	1 month ago	1.27 GB	

3. Discuss

- In your notes, briefly record how you confirmed Docker was installed (e.g., version check, output screenshot).

I confirmed Docker was installed by version check:



A screenshot of a terminal window in a dark-themed code editor. The title bar says 'TutorialClase_Modulo-3'. The tabs at the bottom are 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. The terminal content shows:

```
claudiaMpa@DESKTOP-L5EIQVV MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker --version
Docker version 28.5.2, build ecc6942
```

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 `localhost:<mapped_port>` in your browser.

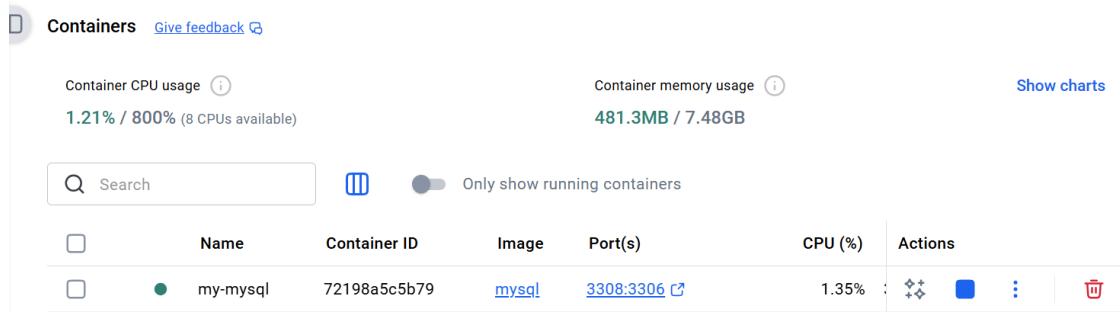
We will follow the teacher's instructions received during the class and only launch one container, in this case the mysql one with the instruction: `docker run --name my-mysql -e MYSQL_ROOT_PASSWORD=secret123 -e MYSQL_DATABASE=appdb -p 3308:3306 mysql`

```

claudiaMphah@DESKTOP-L5EIQVV MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker run --name my-mysql -e MYSQL_ROOT_PASSWORD=secret123 -e MYSQL_DATABASE=appdb -p 3308:3306 mysql
2025-11-23 01:35:00+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.5.0-1.el9 started.
2025-11-23 01:35:00+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-11-23 01:35:00+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.5.0-1.el9 started.
2025-11-23 01:35:00+00:00 [Note] [Entrypoint]: Initializing database files
2025-11-23T01:35:00.877358Z 0 [System] [MY-015017] [Server] MySQL Server Initialization - start.
2025-11-23T01:35:00.879981Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 9.5.0) initializing of server in progress as process 80
2025-11-23T01:35:00.900317Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2025-11-23T01:35:03.879831Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2025-11-23T01:35:09.458213Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
2025-11-23T01:35:26.527933Z 0 [System] [MY-015018] [Server] MySQL Server Initialization - end.
2025-11-23 01:35:26+00:00 [Note] [Entrypoint]: Database files initialized
2025-11-23 01:35:26+00:00 [Note] [Entrypoint]: Starting temporary server
2025-11-23T01:35:26.595605Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2025-11-23T01:35:26.844079Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 9.5.0) starting as process 117
2025-11-23T01:35:26.844114Z 0 [System] [MY-015590] [Server] MySQL Server has access to 8 logical CPUs.
2025-11-23T01:35:26.844134Z 0 [System] [MY-015590] [Server] MySQL Server has access to 8225255424 bytes of physical memory.
2025-11-23T01:35:26.868355Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2025-11-23T01:35:29.539106Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2025-11-23T01:35:30.591787Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2025-11-23T01:35:30.591875Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.

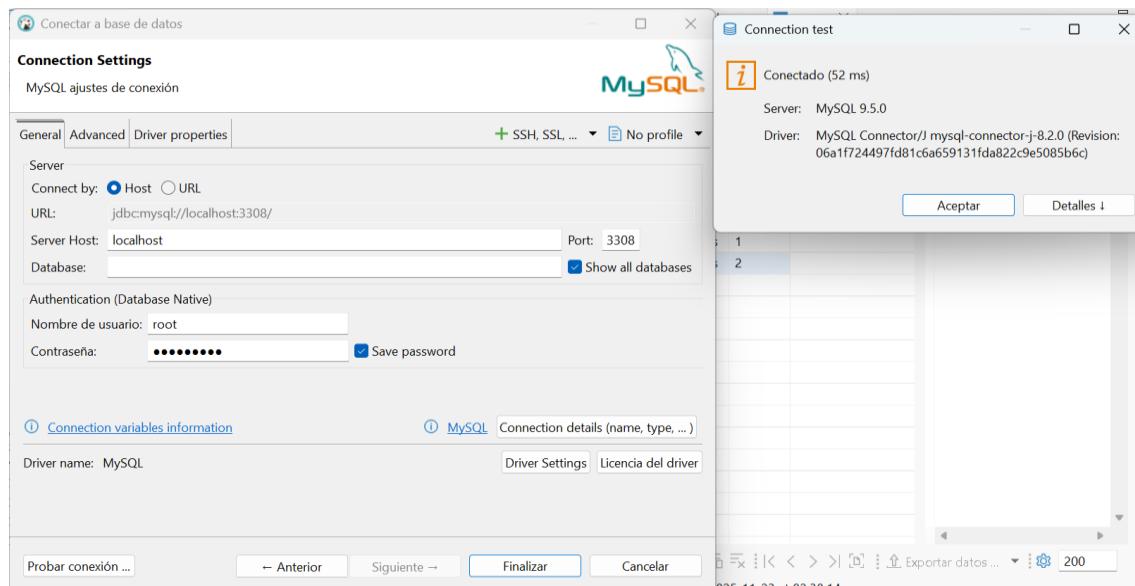
```

Container my-mysql created:

A screenshot of the Docker interface showing the 'Containers' section. It displays resource usage (CPU and memory), a search bar, and a table listing containers. The 'my-mysql' container is listed with the following details: Name: my-mysql, Container ID: 72198a5c5b79, Image: mysql, Port(s): 3308:3306, CPU (%): 1.35%. Actions for the container include a green dot icon, a blue square icon, a three-dot menu icon, and a trash can icon.

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
<input type="checkbox"/>	my-mysql	72198a5c5b79	mysql	3308:3306	1.35%	   

Checking the connection to the MySQL database:

A screenshot of the MySQL Workbench interface. On the left, the 'Connection Settings' dialog shows the following configuration: Connect by: Host, URL: jdbc:mysql://localhost:3308/, Server Host: localhost, Port: 3308, Database: (empty), Authentication (Database Native): Nombre de usuario: root, Contraseña: (redacted), Save password checked. Below this, tabs for 'Connection variables information' and 'MySQL' are visible. On the right, the 'Connection test' dialog shows a successful connection: Conectado (52 ms), Server: MySQL 9.5.0, Driver: MySQL Connector/J mysql-connector-j-8.2.0 (Revision: 06af724497fd81c6a659131fda822c9e5085b6c). Buttons for 'Aceptar' and 'Detalles' are present.

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.

The screenshot shows the Docker UI interface. At the top, there are three status indicators: 'Container CPU usage' (1.21% / 800%, 8 CPUs available), 'Container memory usage' (481.3MB / 7.48GB), and a 'Show charts' button. Below these are search and filter options ('Search' and 'Only show running containers'). A table lists the running container 'my-mysql' with details: Name (my-mysql), Container ID (72198a5c5b79), Image (mysql), Port(s) (3308:3306), CPU (%) (1.35%), and Actions (a dropdown menu and a trash icon). The container is shown with a green dot indicating it is running.

Containers inspect:

The screenshot shows the Docker API inspect endpoint for the 'my-mysql' container. The page includes tabs for Logs, Inspect, Bind mounts, Exec, Files, and Stats. The 'Inspect' tab is selected, showing detailed JSON information about the container. The JSON output is as follows:

```
1 v {
2   "Id": "735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1",
3   "Created": "2025-11-24T15:04:23.257723737Z",
4   "Path": "docker-entrypoint.sh",
5   "Args": [
6     "mysqld"
7   ],
8   "State": {
9     "Status": "running",
10    "Running": true,
11    "Paused": false,
12    "Restarting": false,
13    "OOMKilled": false,
14    "Dead": false,
15    "Pid": 1670,
16    "ExitCode": 0,
17    "Error": "",
18    "StartedAt": "2025-11-24T15:04:26.648730222Z",
19    "FinishedAt": "2001-01-01T00:00:00Z"
20  },
21  "Image": "sha256:569c4128dfa625ac2ac62cd8af588a3a6a60a049da8d8fbfac95880ecdbe5",
22  "ResolvConfPath": "/var/lib/docker/containers/735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1/resolv.conf",
23  "HostnamePath": "/var/lib/docker/containers/735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1/hostname",
24  "HostPath": "/var/lib/docker/containers/735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1/host",
25  "LogPath": "/var/lib/docker/containers/735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1/735cd42616d1a9940e8bbfde9a18944fffb98c3285a4a60a952c14336278902e1.json.log",
26  "Name": "my-mysql",
27  "RestartCount": 0,
28  "Driver": "overlayfs",
29  "Platform": "linux",
30  "MountLabel": "",
31  "ProcessLabel": "",
32  "AppArmorProfile": "",
33  "ExecIDs": [
34    "e1f5d707c351b165eee371493341d7e6b707e116ba2ca2ddaa2fb4ccb75c926e8"
35  ]
36 }
```

Containers Logs:

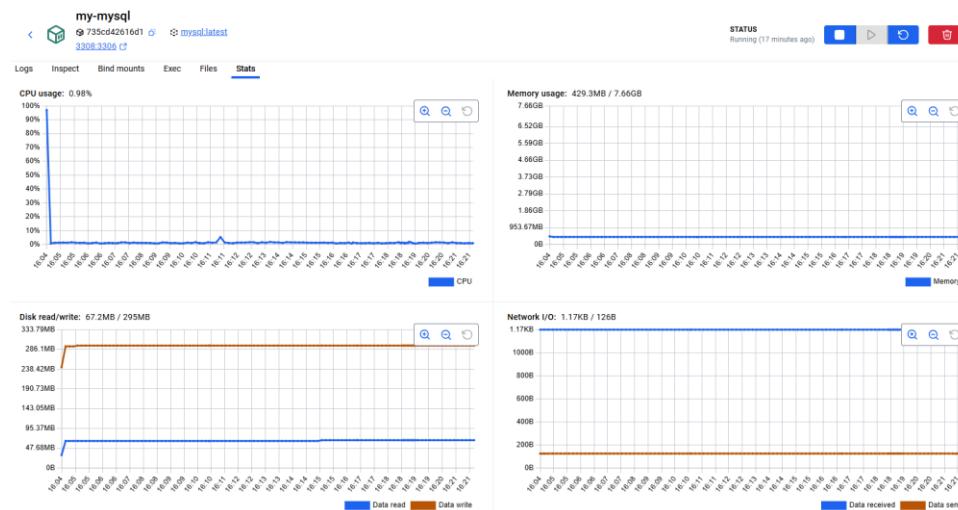
```
my-mysql
735cd42616d1 mysql:latest
3308:3306

Logs Inspect Bind mounts Exec Files Stats STATUS
Running (16 minutes ago) □ ▶ 🔍 🗑️

Logs
Bind mounts
Exec
Files
Stats
```

2025-11-24T15:04:34.058171Z 0 [System] [MV-0159B] [Server] MySQL Server has access to 8225255424 bytes of physical memory.
2025-11-24T15:04:34.083854Z 1 [System] [MV-01356] [InnoDB] Innodb initialization has started.
2025-11-24T15:04:34.088882Z 1 [System] [MV-01357] [InnoDB] Innodb initialization has ended.
2025-11-24T15:04:35.203381Z 0 [Warning] [MV-01068] [Server] CA certificate ca.pem is self signed.
2025-11-24T15:04:35.203437Z 0 [System] [MV-01362] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2025-11-24T15:04:35.209782Z 0 [Warning] [MV-01181] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2025-11-24T15:04:35.209837Z 0 [System] [MV-01323] [Server] X Plugin ready for connections. Socket: /var/run/mysqlx.sock
2025-11-24T15:04:35.209837Z 0 [System] [MV-01323] [Server] X Plugin ready for connections. Socket: /var/run/mysqlx.sock
2025-11-24 15:04:35+00:00 [Note] [Entrypoint]: Temporary server started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysql.sock'
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leapseconds' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/tzdata.ztl' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
2025-11-24 15:04:39+00:00 [Note] [Entrypoint]: Creating database appdb
2025-11-24 15:04:39+00:00 [Note] [Entrypoint]: Stopping temporary server
2025-11-24T15:04:39.634821Z 12 [System] [MV-01317] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 9.5.0).
2025-11-24 15:04:40+00:00 [Note] [Entrypoint]: Temporary server stopped
2025-11-24T15:04:40.643906Z 0 [System] [MV-010910] [Server] /usr/sbin/mysqld. Shutdown complete (mysqld 9.5.0) MySQL Community Server - GPL.
2025-11-24T15:04:40.643962Z 0 [System] [MV-015916] [Server] MySQL Server - end.
2025-11-24 15:04:40+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.

Containers Stats:



Containers Exec:

```
Containers / my-mysql
my-mysql
735cd42616d1 mysql:latest
3308:3306

Logs Inspect Bind mounts Exec Files Stats STATUS
Running (
```

Logs

Bind mounts

Exec

Files

Stats

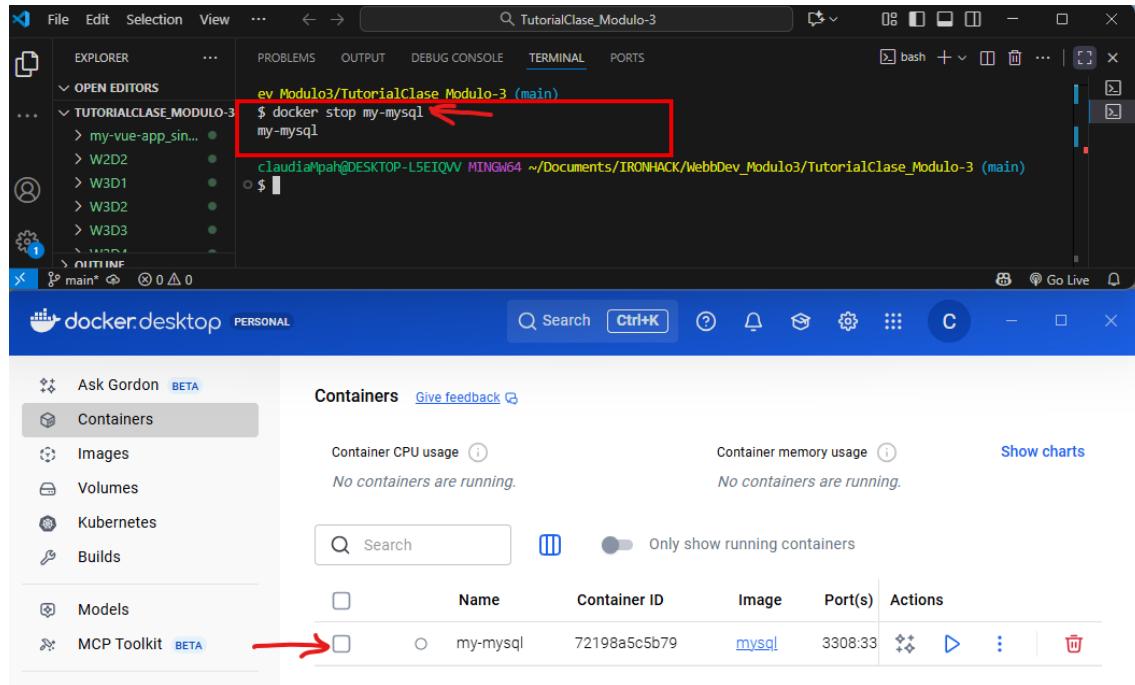
```
sh-5.1# ls
adm cache db empty ftp games kerberos lib local lock log mail nis opt preserve run spool tmp yp
sh-5.1# cd lib/mysql
sh-5.1# ls
'ib_16384_0_dblwr' '#innodb_temp' binlog.000001 ca-key.pem client-key.pem ibtmp1 mysql.sock private_key.pem server-key.pem undo_002
'ib_16384_1_dblwr' appdb binlog.000002 ca.pem ib_buffer_pool mysql mysql_upgrade_history public_key.pem sys undo_001
'innodb redo' auto.cnf binlog.index client-cert.pem ibdata1 mysql.ibd performance_schema server-cert.pem
sh-5.1#
```

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).

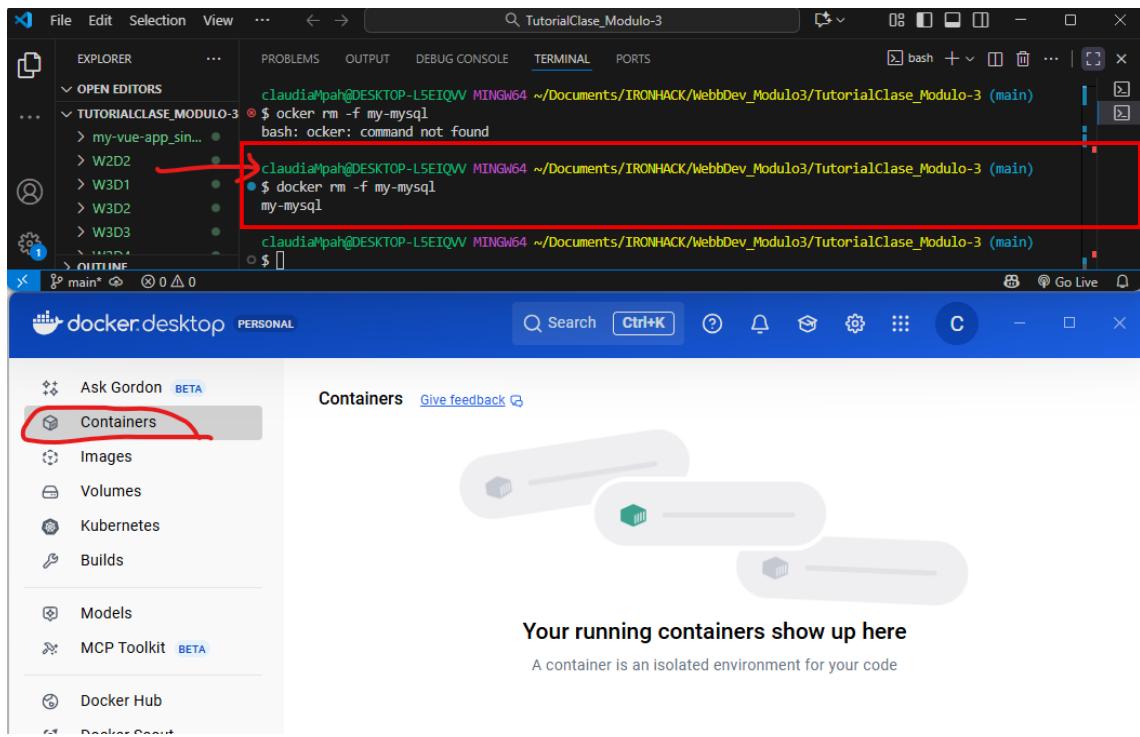
When we stop the container from the console, we see that it also stops from the Docker application.



2. Remove

- Remove both containers now that they're stopped.
- Verify they no longer appear in your container list.

When we delete the container from the console, we see how it is deleted from the Docker application.



3. Clean Up Images

- If you won't reuse the images, remove them to free disk space.
- Confirm your local image list is updated accordingly.

We've noticed that removing the container doesn't remove the image. Using the command "**docker rmi mysql**" from the console successfully removed the image from Docker.

The screenshot shows a terminal window in a code editor (VS Code) and the Docker Desktop application.

Terminal (VS Code):

```
claudiaMpah@DESKTOP-L5EIQVW MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker images
claudiaMpah@DESKTOP-L5EIQVW MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ docker rmi mysql
Untagged: mysql:latest
Deleted: sha256:569c4128dfa625ac2ac62cdd8af588a3a6a60a049d1a8d8f0fac95880ecdbbe5
claudiaMpah@DESKTOP-L5EIQVW MINGW64 ~/Documents/IRONHACK/WebbDev_Modulo3/TutorialClase_Modulo-3 (main)
$ |
```

Docker Desktop:

- Left sidebar:
 - Ask Gordon BETA
 - Containers
 - Images** (highlighted with a red circle)
 - Volumes
 - Kubernetes
 - Builds
 - Models
 - MCP Toolkit BETA
- Center:
 - Images Give feedback
 - Local My Hub
 - Diagram showing the relationship between code, Dockerfile, images, and containers.
 - Images are used to run containers**
 - Search bar: Search images to run