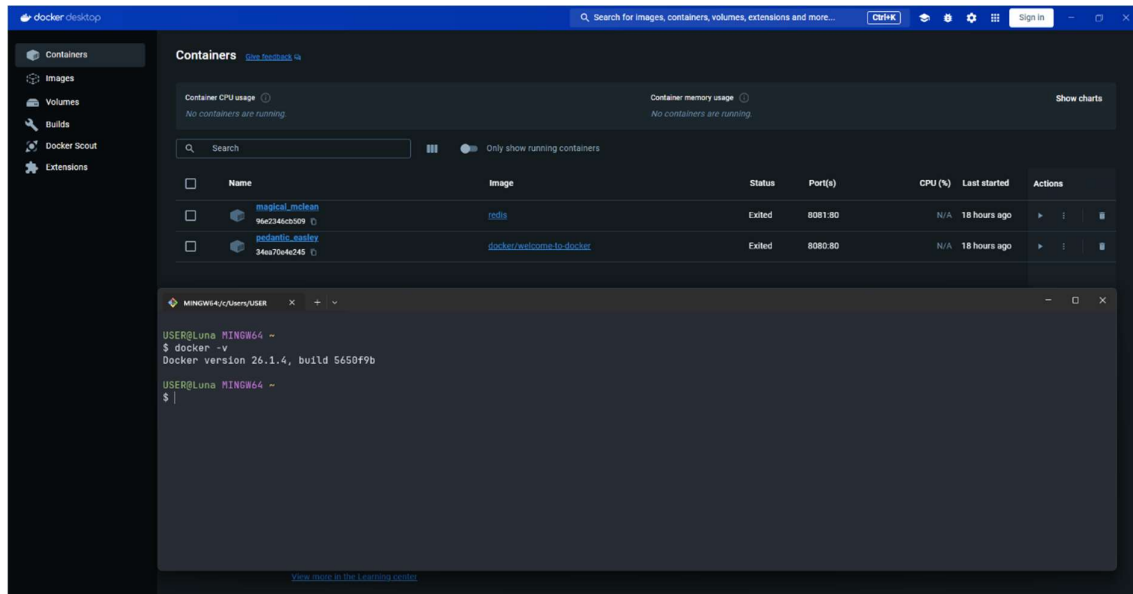


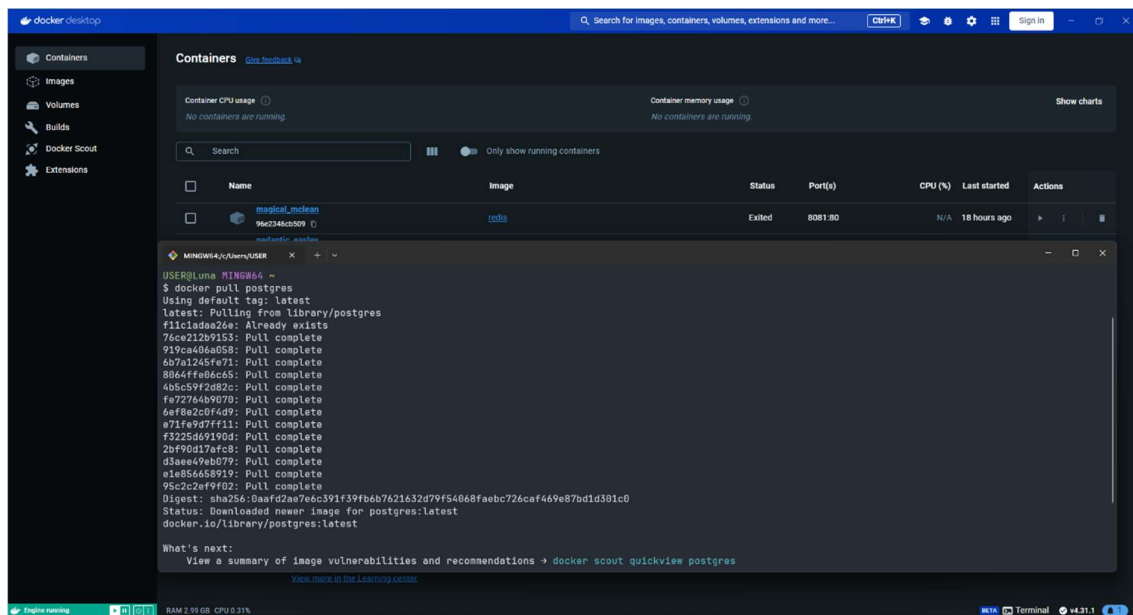
# Docker

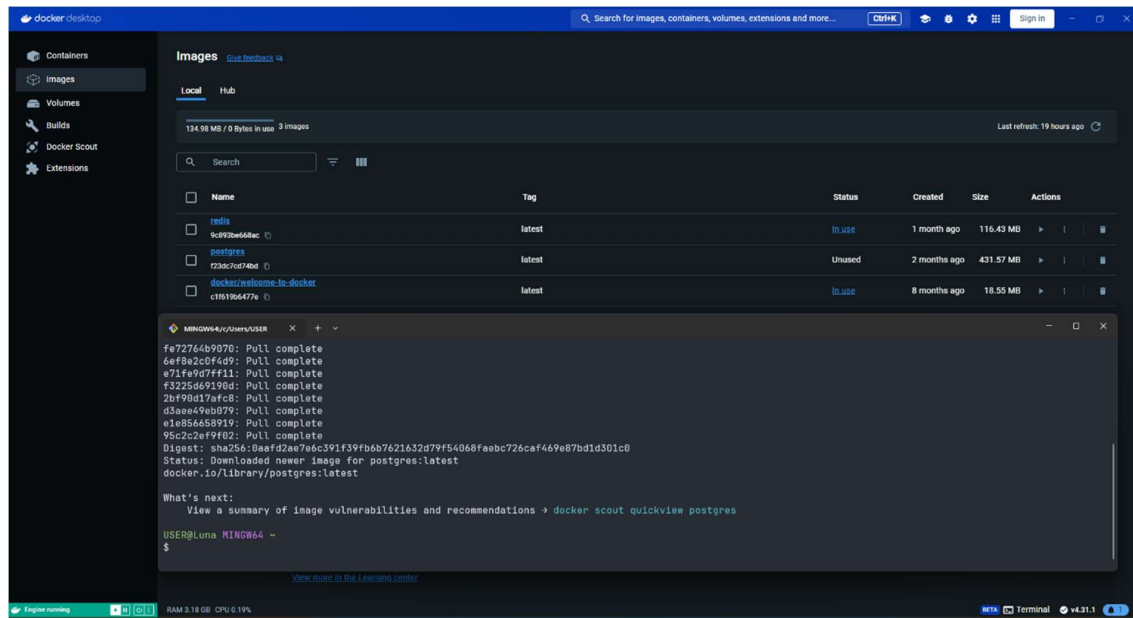
## PART 1 - Basic Command Docker

### 1. Docker in your local computer \$ **docker -version**

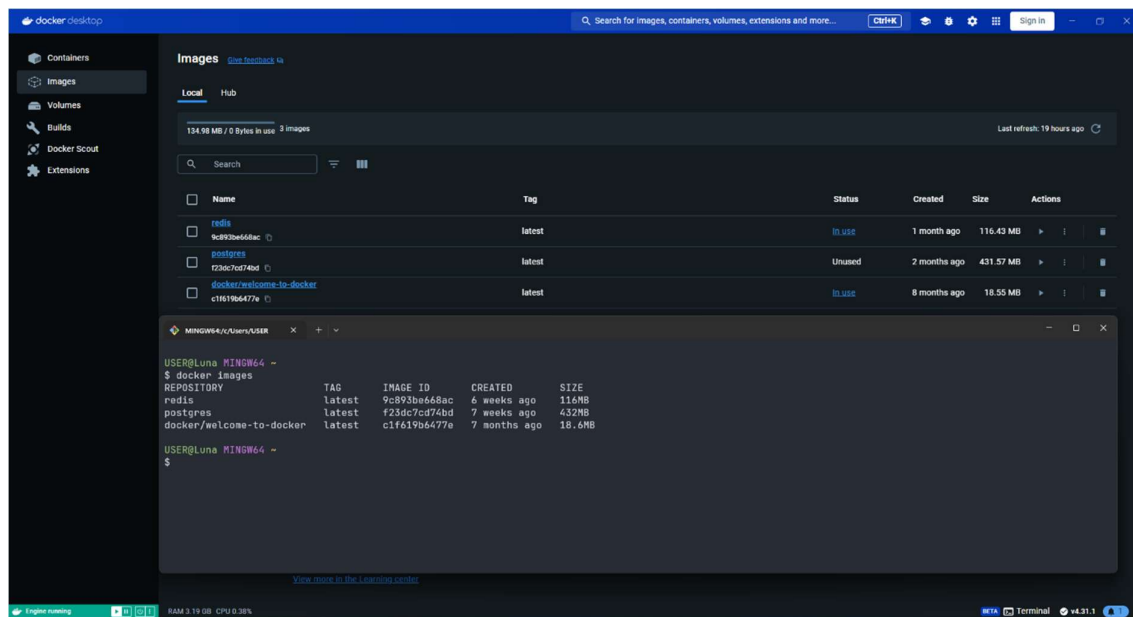


### 2. Pull an image Postgres from a Docker Hub with command \$ **docker pull**

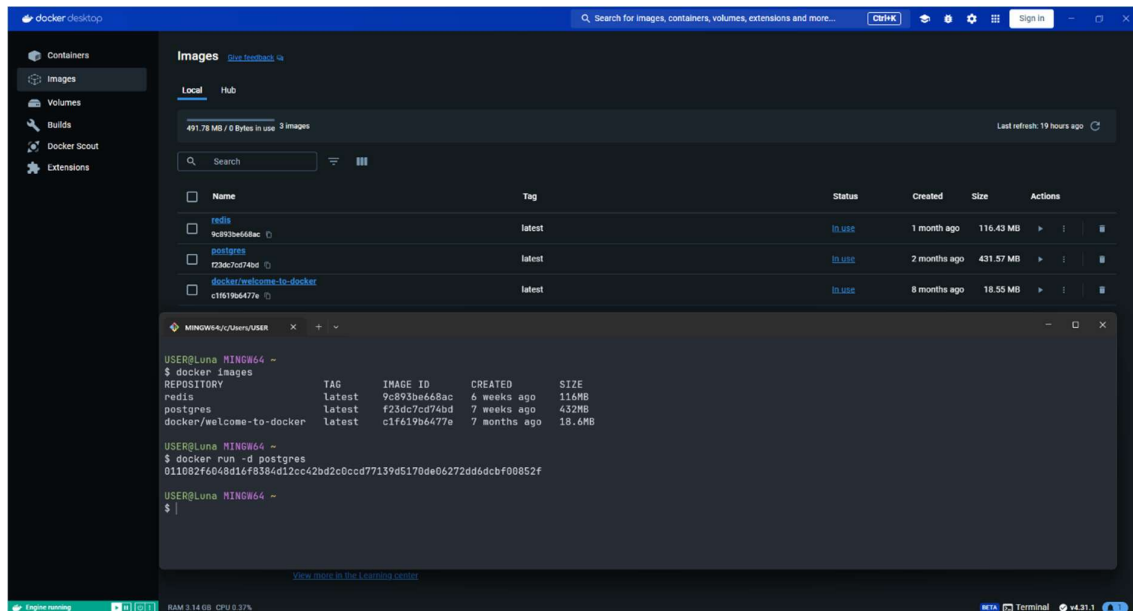




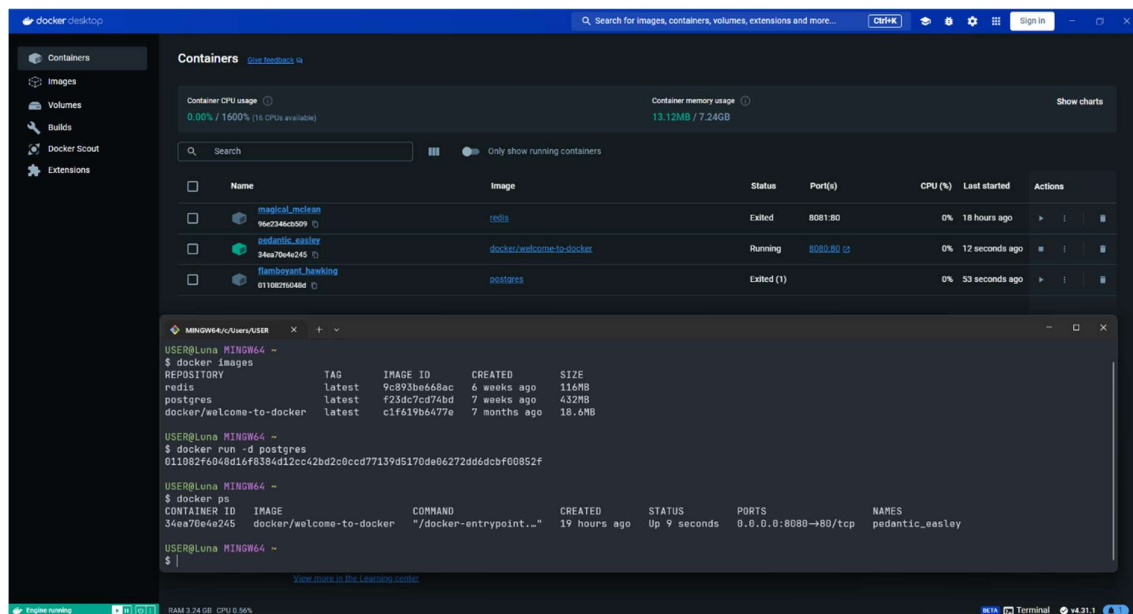
### 3. List local images with command `$ docker images`



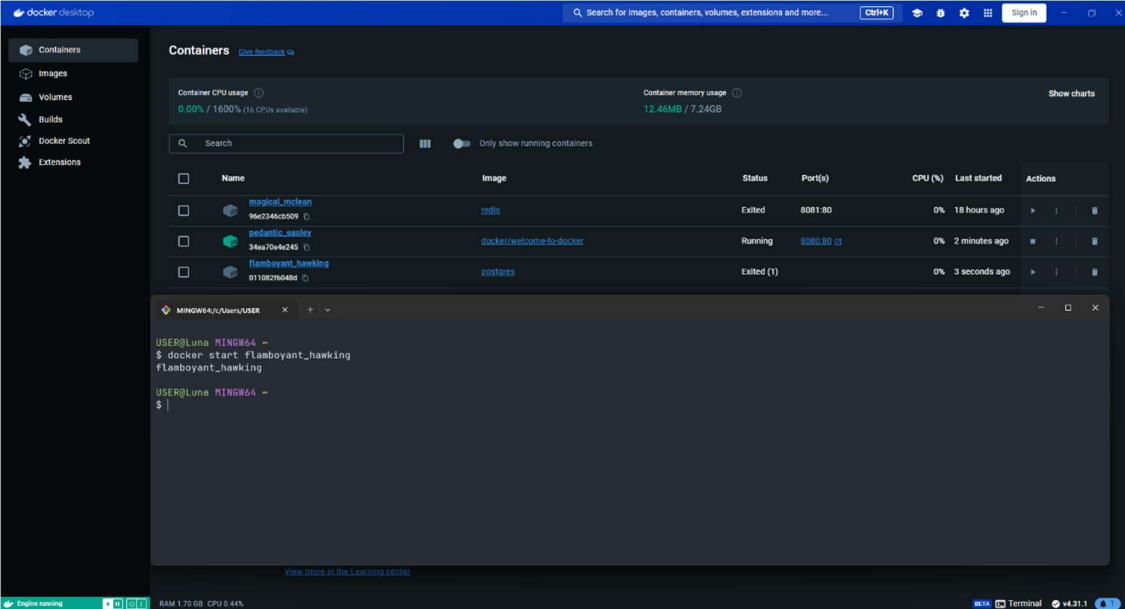
4. Create and run a container from an image with command `$ docker run -d <image-name>`



5. List currently running containers with command `$ docker ps`



6. Start and stop an existing container with command `$ docker start|stop <container_name>`  
Start an exist container

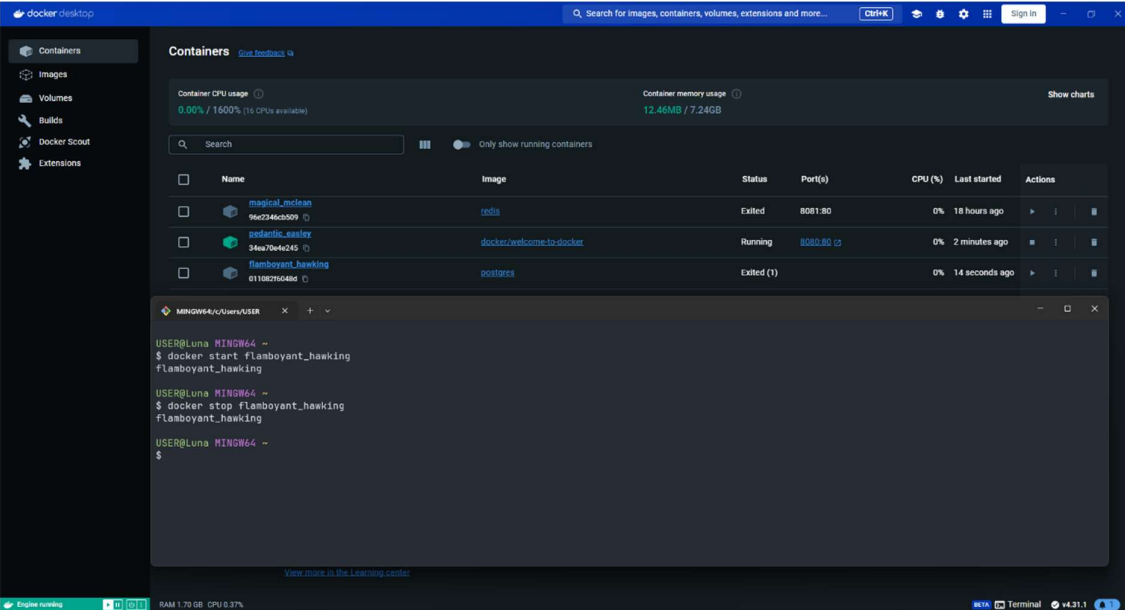


The screenshot shows the Docker Desktop interface. The 'Containers' tab is active, displaying a table of containers. The 'flamboyant\_hawking' container is running. A terminal window is open, showing the command `docker start flamboyant_hawking` being executed.

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
magical_mclean	redis	Exited	8081:80	0%	18 hours ago	
pedantic_sasley	docker/welcome-to-docker	Running	8080:80	0%	2 minutes ago	
flamboyant_hawking	postgres	Exited (1)		0%	3 seconds ago	

```
USER@Luna MINGW64 -  
$ docker start flamboyant_hawking  
flamboyant_hawking  
  
USER@Luna MINGW64 -  
$
```

## Stop an exist container



The screenshot shows the Docker Desktop interface. The 'Containers' tab is active, displaying a table of containers. The 'flamboyant\_hawking' container is running. A terminal window is open, showing the command `docker stop flamboyant_hawking` being executed.

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
magical_mclean	redis	Exited	8081:80	0%	18 hours ago	
pedantic_sasley	docker/welcome-to-docker	Running	8080:80	0%	2 minutes ago	
flamboyant_hawking	postgres	Exited (1)		0%	14 seconds ago	

```
USER@Luna MINGW64 -  
$ docker start flamboyant_hawking  
flamboyant_hawking  
  
USER@Luna MINGW64 -  
$ docker stop flamboyant_hawking  
flamboyant_hawking  
  
USER@Luna MINGW64 -  
$
```

7. Run a container with and publish a container's port(s) to the host with command `$ docker run -p <host_port>:<container_port> -d <image_name>`

The screenshot shows the Docker Desktop interface. At the top, there are statistics for container CPU usage (0.00% / 1600%) and container memory usage (12.46MB / 7.24GB). Below this is a table of containers:

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
magical_mclean	redis	Exited	8081:80	0%	18 hours ago	[stop] [refresh] [logs]
pedantic_easley	docker/welcome-to-docker	Running	8080:80	0%	4 minutes ago	[stop] [refresh] [logs]
flamboyant_hawking	postgres	Exited (1)		0%	2 minutes ago	[stop] [refresh] [logs]
reverent_devind	postgres	Exited (1)	8082:80	0%	50 seconds ago	[stop] [refresh] [logs]

Below the table is a terminal window with the following commands and output:

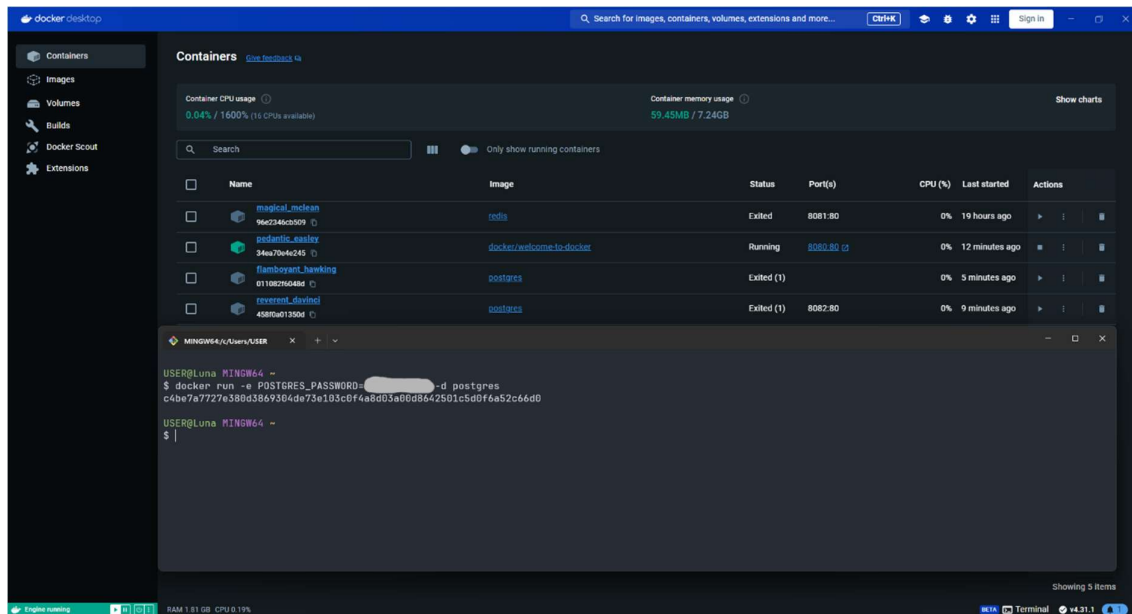
```
USER@Luna MINGW64 ~  
$ docker start flamboyant_hawking  
flamboyant_hawking  
  
USER@Luna MINGW64 ~  
$ docker stop flamboyant_hawking  
flamboyant_hawking  
  
USER@Luna MINGW64 ~  
$ docker run -p 8082:80 -d postgres  
458f6m01350dd6f375fe6bfce7381a2d9a39fbfd79a2b56c70daa0cda323b0  
  
USER@Luna MINGW64 ~  
$
```

8. Fetch and follow the logs of a container with command `$ docker logs -f <container_name>`

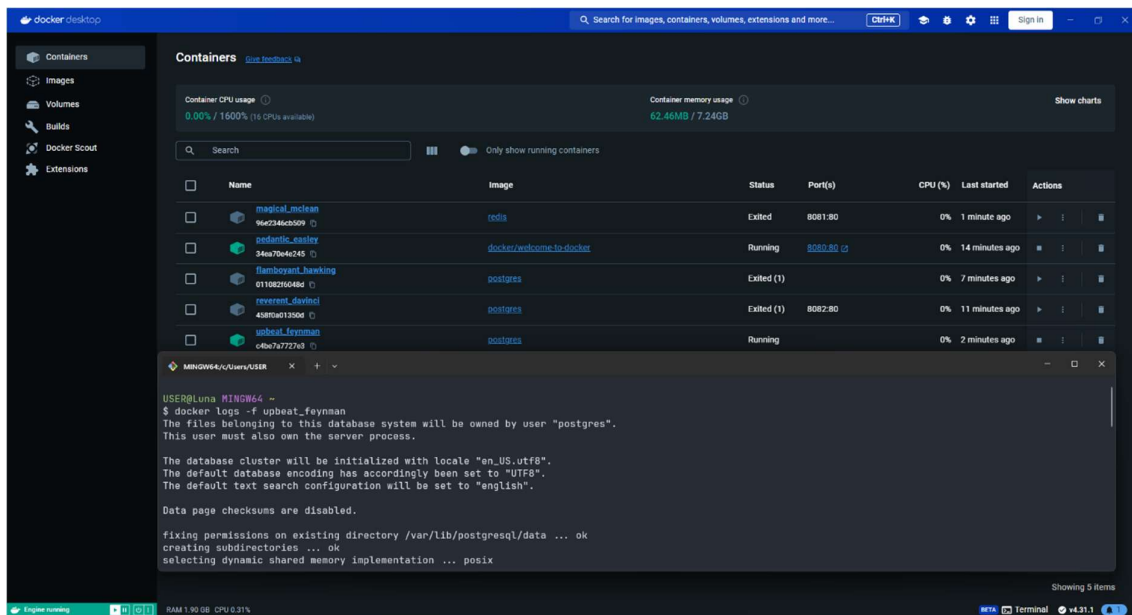
The screenshot shows the Docker Desktop interface, similar to the previous one. The container table is the same. The terminal window now shows the output of the `docker logs -f pedantic_easley` command:

```
USER@Luna MINGW64 ~  
$ docker logs -f pedantic_easley  
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration  
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/  
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh  
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf  
/docker-entrypoint.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf  
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh  
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh  
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh  
/docker-entrypoint.sh: Configuration complete; ready for start up  
2024/07/03 13:05:34 [notice] 1#1: using the "epoll" event method  
2024/07/03 13:05:34 [notice] 1#1: nginx/1.25.1  
2024/07/03 13:05:34 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1.git20220924-r10)  
2024/07/03 13:05:34 [notice] 1#1: OS: Linux 5.15.153.1-microsoft-standard-WSL2  
2024/07/03 13:05:34 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576  
  
View more in the Learning center
```

Jika belum terhubung dengan postgres, sambungkan dulu dengan command `$ docker run -e POSTGRES_PASSWORD=password -d postgres` untuk menyambungkannya



Jika sudah terhubung, kita bisa mengakses logs container dengan image postgres



9. Open a shell inside a running container with command `$ docker exec -it <container_name> sh`

The screenshot shows the Docker Desktop application. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Docker Scout, and Extensions. The main area displays the 'Containers' tab with a search bar and a toggle for 'Only show running containers'. Below this is a table of containers:

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
magical_mclean	redis	Exited	8081:80	0%	7 minutes ago	[Restart] [Stop] [Logs]
padanite_easley	docker/welcome-to-docker	Running	8080:80	0%	19 minutes ago	[Restart] [Stop] [Logs]
flamboyant_hawking	postgres	Exited (1)		0%	13 minutes ago	[Restart] [Stop] [Logs]
reverent_devinci	postgres	Exited (1)	8082:80	0%	16 minutes ago	[Restart] [Stop] [Logs]
upbeat_feynman	postgres	Running		0%	8 minutes ago	[Restart] [Stop] [Logs]

Below the table, a terminal window titled 'MINGW64 PowerShell' is open, showing the command `$ docker exec -it upbeat_feynman sh` and the output of `postgres --version`:

```
USER@Luna MINGW64 -  
$ docker exec -it upbeat_feynman sh  
# postgres --version  
postgres (PostgreSQL) 16.3 (Debian 16.3-1.pgd120+1)  
#
```