

School: CLOUDBOOSTA	Course: DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Lab-2_ Docker-1
Student No: CBALC0168	Week: 4 of 8
	Issue Date: May 23, 2026

Tasks:

1. Create Docker Hub repository
2. Configure Docker Hub repository visibility
3. Login to Docker Hub using Docker CLI
4. Verify Docker Hub authentication
5. Configure Docker container image tagging
6. Configure Dockerfile Apache container definition
7. Build Docker container image
8. Verify Docker image build process
9. Verify Docker image repository listing
10. Run Apache Docker container instance
11. Configure Docker port mapping
12. Verify running Docker container status
13. Access Apache containerized web server
14. Verify Apache Ubuntu default page deployment
15. Push Docker image to Docker Hub repository
16. Verify Docker image push execution
17. Verify Docker Hub repository image upload
18. Configure Docker ENTRYPOINT demonstration container
19. Build sleepvsepoint Docker image
20. Execute Docker sleep container instances
21. Verify Docker container execution lifecycle
22. Verify Docker exited and running container states
23. Validate Docker image management workflow
24. Validate Docker Hub cloud repository workflow

Goal:

To configure Docker Hub cloud repository management, Docker CLI authentication, Docker image creation, Apache container deployment, Docker image publishing, Docker container lifecycle management, ENTRYPOINT container execution workflows, and cloud-native Docker image orchestration within AWS EC2 and Docker Hub infrastructure.

Result:

- Docker Hub repository configured successfully
- Docker CLI authentication configured successfully
- Docker image tagging configured successfully
- Apache Docker container image configured successfully
- Docker image build process executed successfully
- Docker image repository listing validated successfully
- Apache container deployment configured successfully
- Docker port mapping configured successfully
- Apache Ubuntu default page deployed successfully
- Docker image pushed to Docker Hub successfully
- Docker Hub repository image upload validated successfully
- Docker ENTRYPOINT container workflow configured successfully
- Sleep container image built successfully
- Docker container lifecycle execution validated successfully
- Docker running and exited container states validated successfully
- Docker Hub cloud-native workflow implemented successfully

Not secure 18.196.192.170:8080



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.Load  
|   |-- *.conf
```

Action:

Step 1: Created Docker Hub Repository

- Opened: Docker Hub → Repositories → Create
- Configured:
 - Repository name: docker_demo
 - Repository visibility: Public
- Verified Docker Hub repository created successfully

Step 2: Configured Docker Hub Repository Description

- Configured Docker repository description successfully
- Verified Docker Hub repository metadata configured successfully

Step 3: Logged into Docker Hub Using Docker CLI

- Opened Ubuntu EC2 terminal successfully
- Executed:
 - `docker login -u samuelakosaonyekwe`
- Entered Docker Hub authentication password successfully
- Verified Docker CLI authentication completed successfully

Step 4: Verified Docker Hub Authentication Status

- Verified Docker login success message successfully
- Confirmed Docker Hub CLI authentication functioning successfully

Step 5: Configured Apache Dockerfile Container Definition

- Opened: Dockerfile
- Configured:
 - Ubuntu 18.04 base image
 - Apache2 package installation
 - Port exposure configuration

- Apache foreground execution command
- Verified Dockerfile container definition configured successfully

Step 6: Built Docker Container Image

- Executed:
 - `docker build -t samuelakosaonyejekwe/docker_demo:v1 .`
- Verified Docker build process initialized successfully
- Confirmed Docker image exported successfully

Step 7: Verified Docker Image Build Process

- Verified:
 - Docker build layers
 - Package installation execution
 - Docker image unpacking process
- Confirmed Docker image build completed successfully

Step 8: Verified Docker Image Repository Listing

- Executed:
 - `docker images`
- Verified Docker images:
 - `myapp:latest`
 - `samuelakosaonyejekwe/docker_demo:v1`
- Confirmed Docker image repository listing validated successfully

Step 9: Executed Apache Docker Container Deployment

- Executed Docker container run command successfully
- Configured:
 - Apache container image
 - Port mapping: 8080:80
 - Container name: my-container

- Verified Apache Docker container deployed successfully

Step 10: Verified Running Docker Containers

- Executed:
 - `docker ps`
- Verified:
 - Running container ID
 - Apache container image
 - Port mappings
 - Container uptime
- Confirmed Docker running container status validated successfully

Step 11: Accessed Apache Containerized Web Server

- Opened browser using EC2 public IP and port 8080 successfully
- Verified Apache Ubuntu default page accessible successfully

Step 12: Verified Apache Ubuntu Default Page Deployment

- Verified:
 - Apache2 Ubuntu Default Page
 - Apache web server functionality
 - Ubuntu Apache configuration overview
- Confirmed Apache web server deployment functioning successfully

Step 13: Pushed Docker Image to Docker Hub Repository

- Executed:
 - `docker push samuelakosaonyejekwe/docker_demo:v1`
- Verified Docker image layer upload completed successfully
- Confirmed Docker image push execution completed successfully

Step 14: Verified Docker Hub Repository Upload

- Opened Docker Hub repository successfully
- Verified:
 - Repository image tag: v1
 - Docker image upload timestamp
 - Docker image repository overview
- Confirmed Docker Hub image repository upload validated successfully

Step 15: Configured Docker ENTRYPOINT Demonstration Container

- Opened: Dockerfile
- Configured:
 - Alpine Linux base image
 - Sleep execution ENTRYPOINT workflow
- Verified Docker ENTRYPOINT container definition configured successfully

Step 16: Built sleepvsepoint Docker Image

- Executed:
 - `docker build -t sleepvsepoint:latest .`
- Verified Docker build process completed successfully
- Confirmed sleepvsepoint Docker image created successfully

Step 17: Verified sleepvsepoint Docker Image Repository

- Executed:
 - `docker images`
- Verified Docker image:
 - `sleepvsepoint:latest`
- Confirmed sleepvsepoint image repository validated successfully

Step 18: Executed Docker Sleep Container Instance

- Executed:
 - `docker run -d --name example sleepvsepoint:latest`
- Verified sleep container started successfully
- Confirmed Docker detached container execution functioning successfully

Step 19: Verified Docker Container Lifecycle Execution

- Executed:
 - `docker ps -a`
- Verified:
 - Running container states
 - Exited container states
 - Sleep command execution lifecycle
- Confirmed Docker container lifecycle validated successfully

Step 20: Executed Docker Sleep Container with Runtime Argument

- Executed:
 - `docker run -d --name example2 sleepvsepoint:latest 30`
- Verified Docker runtime argument execution successfully
- Confirmed container sleep duration configured successfully

Step 21: Verified Running and Exited Docker Containers

- Executed:
 - `docker ps -a`
- Verified:
 - example container exited successfully
 - example2 container running successfully
 - Apache container running successfully
- Confirmed Docker container execution states validated successfully

Step 22: Verified Docker Container Management Workflow

- Verified:
 - Docker image build workflow
 - Docker image tagging workflow
 - Docker Hub image publishing workflow
 - Docker runtime lifecycle workflow
- Confirmed Docker container management functioning successfully

Step 23: Verified Docker Hub Cloud Repository Workflow

- Verified:
 - Docker cloud repository accessibility
 - Docker image version management
 - Docker Hub image synchronization
 - Docker CLI cloud integration
- Confirmed Docker Hub cloud-native workflow validated successfully

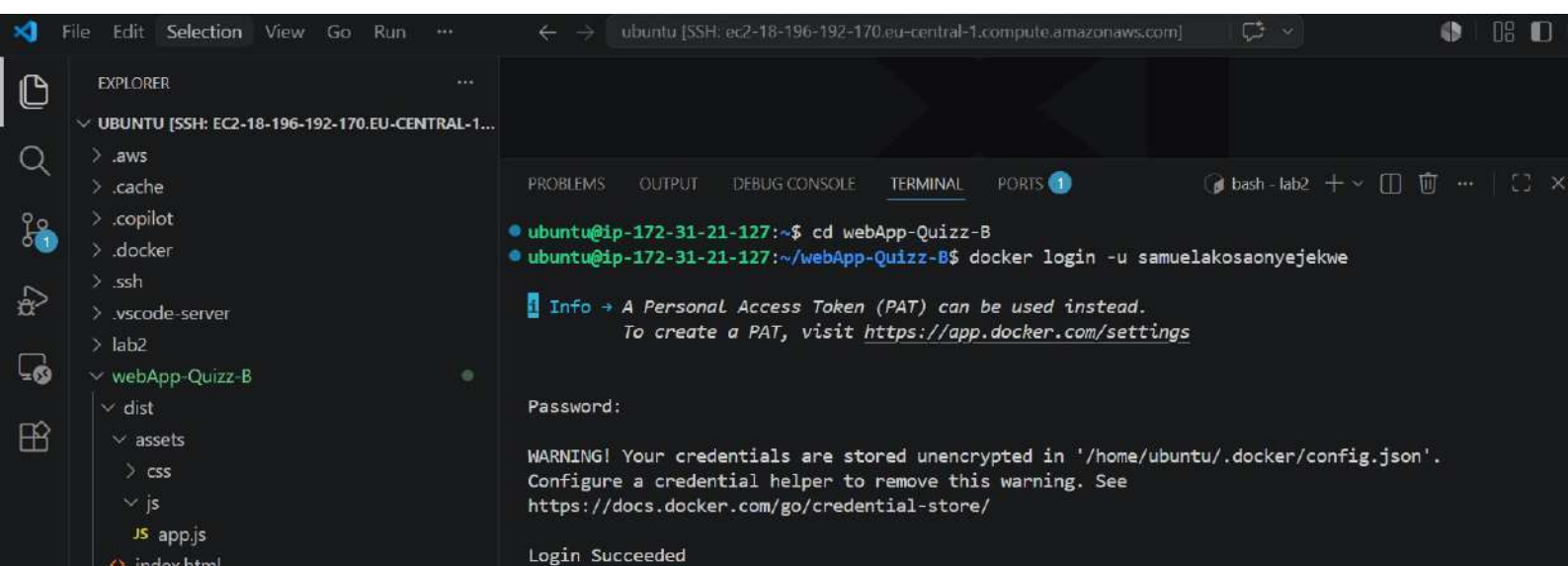
Summary

- Docker Hub repository configured successfully
- Docker CLI authentication validated successfully
- Apache Docker container deployment configured successfully
- Docker image build and tagging workflow validated successfully
- Docker image push execution validated successfully
- Docker Hub image repository upload validated successfully
- Docker ENTRYPOINT workflow validated successfully
- Docker container lifecycle management validated successfully
- Apache web server container deployment validated successfully
- Docker Hub cloud-native image orchestration implemented successfully

Lab Objective Achieved:

Implemented Docker Hub repository management, Docker CLI authentication, Apache Docker container deployment, Docker image creation and publishing, Docker image tagging workflows, Docker Hub cloud synchronization, Docker ENTRYPOINT lifecycle management, Docker runtime execution workflows, Apache containerized web server hosting, and cloud-native

Docker container orchestration successfully within AWS EC2 and Docker Hub cloud infrastructure.



File Edit Selection View Go Run ... ubuntu [SSH: ec2-18-196-192-170.eu-central-1.compute.amazonaws.com]

EXPLORER

- UBUNTU [SSH: EC2-18-196-192-170.EU-CENTRAL-1...
- .aws
- .cache
- .copilot
- .docker
- .ssh
- .vscode-server
- lab2
 - Dockerfile
- webApp-Quizz-B
 - dist
 - assets
 - css
 - js
 - app.js
 - index.html
 - .gitignore
 - Dockerfile
 - package.json
 - pnpm-lock.yaml
 - server.js
 - .bash_history
 - .bash_logout
 - .bashrc
 - OUTLINE
 - TIMELINE

Dockerfile

```
1 FROM ubuntu:18.04
2 LABEL maintainer="my-name"
3 LABEL remarks="This is a dockerfile"
4 RUN apt-get update && apt-get install apache2 -y && apt-get clean
5 EXPOSE 8080
6 CMD ["apache2ctl", "-DFOREGROUND"]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - lab2

```
ubuntu@ip-172-31-21-127:~$ mkdir lab2
ubuntu@ip-172-31-21-127:~$ cd lab2/
ubuntu@ip-172-31-21-127:~/lab2$ docker build -t samuelakosaonyejekwe/docker_demo:v1 .
[+] Building 27.1s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 233B
=> [internal] load metadata for docker.io/library/ubuntu:18.04
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/2] FROM docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697
=> => resolve docker.io/library/ubuntu:18.04@sha256:152dc042452c496007f07ca9127571cb9c29697
=> => sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef 25.69MB / 25.69MB
=> => extracting sha256:7c457f213c7634afb95a0fb2410a74b7b5bc0ba527033362c240c7a11bef4331
=> [2/2] RUN apt-get update && apt-get install apache2 -y && apt-get clean
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:21149a69642658dfb89a74b8a969e637403828c1157d87bf4c8a12fde0b
=> => exporting config sha256:107cfa2a0a673b383363f3464d9d3f03ae50ca100faf1f896f9e64de952f6
=> => exporting attestation manifest sha256:f972d2a1ba86bf064e41cc3a10a3325da65fa1283540c91
=> => exporting manifest list sha256:09a965eed2d418f06274743999aa4b1f9990d7b5e32eb95b491e03
=> => naming to docker.io/samuelakosaonyejekwe/docker_demo:v1
=> => unpacking to docker.io/samuelakosaonyejekwe/docker_demo:v1
```

Ln 6, Col 35 Spaces: 4 UTF-

File Edit Selection View Go Run ... ubuntu [SSH: ec2-18-196-192-170.eu-central-1.compute.amazonaws.com]

EXPLORER

- UBUNTU [SSH: EC2-18-196-192-170.EU-CENTRAL-1...
- > .aws
- > .cache
- > .copilot
- > .docker
- > .ssh
- > .vscode-server
- lab2
 - Dockerfile
 - webApp-Quizz-B
 - dist
 - assets
 - css
 - js
 - app.js
 - index.html
 - .gitignore
 - Dockerfile
 - package.json
 - pnpm-lock.yaml
 - server.js
 - .bash_history
 - .bash_logout
 - .bashrc
 - OUTLINE
 - TIMELINE

Dockerfile

```
lab2 > Dockerfile
1 FROM ubuntu:18.04
2 LABEL maintainer="my-name"
3 LABEL remarks="This is a dockerfile"
4 RUN apt-get update && apt-get install apache2 -y && apt-get clean
5 EXPOSE 8080
6 CMD ["apache2ctl", "-DFOREGROUND"]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

bash - lab2

```
ubuntu@ip-172-31-21-127:~/lab2$ docker images
```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
myapp:latest	22141faa4ca1	1.36GB	352MB	
samuelakosaonyejekwe/docker_demo:v1	09a965eed2d4	312MB	89.1MB	

```
ubuntu@ip-172-31-21-127:~/lab2$ docker run -d -p 8080:80 --name my-container samuelakosaonyejekwe/docker_demo:v1
a65a72404f830dad9d8fe2ccdcba9782b8ea80a45e7cd2f805b5a6348cd74d2
```

```
ubuntu@ip-172-31-21-127:~/lab2$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS
a65a72404f83	samuelakosaonyejekwe/docker_demo:v1	"apache2ctl -DFOREGROUND"	my-container	33 seconds ago	Up 3

```
ubuntu@ip-172-31-21-127:~/lab2$
```

SSH: ec2-18-196-192-170.eu-central-1.com... main* 0 0 1 Ln 6, Col 35 Spaces: 4 UTF-8



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.Load  
|   |-- *.conf
```

```
● ubuntu@ip-172-31-21-127:~/lab2$ docker push samuelakosaonyejekwe/docker_demo:v1
The push refers to repository [docker.io/samuelakosaonyejekwe/docker_demo]
86961d5dbce5: Pushed
7c457f213c76: Pushed
f9a1daa043c2: Pushed
v1: digest: sha256:09a965eed2d418f06274743999aa4b1f9990d7b5e32eb95b491e03c8cac49af3 size: 855
❖ ubuntu@ip-172-31-21-127:~/lab2$
```

hub

Explore

My Hub

Search Docker Hub

CtrlK

S

🏠

samuelakosaonyekwe...

Your personal account

Repositories

Hardened Images

Collaborations

Settings

Default privacy

Notifications

Billing

Usage

Pulls

Storage

Repositories / docker_demo / General

samuelakosaonyekwe/docker_demo

Last pushed 2 minutes ago · Repository size: 85 MB · ☆ 0 · ↓ 0

Docker demo repository

Add a category

General

Tags

Image Management

Collaborators

Webhooks

Settings

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	less than 1 day	2 minutes

See all

Repository overview

An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

Add overview

Docker commands

To push a new tag to this repository:

```
docker push samuelakosaonyekwe/docker_demo:tagname
```

Public view

buildcloud

Build with Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

Go to Docker Build Cloud

Upgrade plan

docker

Explore

Containers

Pricing

Account

Billing

Resources

Blog

Download Docker

Support

Feedback

Documentation

Hub Release Notes

Forums

Company

About Us

Resources

Blog

Customers

Partners

Newsroom

Events and Webinars

Careers

Contact Us

System Status

© 2026 Docker, Inc. All rights reserved. | Terms of Service | Subscription Service Agreement | Privacy | Legal

Facebook

Twitter

YouTube

LinkedIn

RSS

Customize Settings

File Edit Selection View Go Run ... ubuntu [SSH: ec2-18-196-192-170.eu-central-1.compute.amazonaws.com]

EXPLORER

- UBUNTU [SSH: EC2-18-196-192-170.EU-CENTRAL-1...
- > .aws
- > .cache
- > .copilot
- > .docker
- > .ssh
- > .vscode-server
- > lab2
- lab2-task3
 - Dockerfile
- webApp-Quizz-B
 - dist
 - assets
 - css
 - js
 - app.js
 - index.html
 - .gitignore
 - Dockerfile
 - package.json
 - pnpm-lock.yaml
 - server.js

Dockerfile

```
lab2-task3 > Dockerfile
2 LABEL maintainer="my-name"
3 LABEL remarks="This is a dockerfile to check the difference between entrypoint & cd"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

bash - lab2-task3

```
ubuntu@ip-172-31-21-127:~$ mkdir lab2-task3
ubuntu@ip-172-31-21-127:~$ cd lab2-task3
ubuntu@ip-172-31-21-127:~/lab2-task3$ docker build -t sleepvsepoint:latest .
[+] Building 1.9s (6/6) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 194B                                              0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                  1.1s
=> [auth] library/alpine:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/1] FROM docker.io/library/alpine:latest@sha256:5b10f432ef3da1b8d4c7eb6c487f2f5a8f096b 0.3s
=> => resolve docker.io/library/alpine:latest@sha256:5b10f432ef3da1b8d4c7eb6c487f2f5a8f096b 0.0s
=> => sha256:6a0ac1617861a677b045b7ff88545213ec31c0ff08763195a70a4a5adda577 3.86MB / 3.86MB 0.2s
=> exporting to image                                                            0.5s
=> => exporting layers                                                            0.0s
=> => exporting manifest sha256:492276683932b9e88630529f94c5d2cd29d4ee8fdb7c17c5d31ebf76c1b 0.0s
=> => exporting config sha256:dca84a7b4919f1d1fe612db9f26512abe36ee8df0b7d1a421462e0605a2d8 0.0s
=> => exporting attestation manifest sha256:beab2d77fcb951f3fda0b0746f6b4a344c82a8159421a84 0.0s
=> => exporting manifest list sha256:64f42ce38f89d01951be4b4f735019a27080196d048b297b69e6e6 0.0s
=> => naming to docker.io/library/sleepvsepoint:latest                         0.0s
=> => unpacking to docker.io/library/sleepvsepoint:latest                       0.4s
```

```

● ubuntu@ip-172-31-21-127:~/lab2-task3$ docker images

```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA	In Use
myapp:latest	22141faa4ca1	1.36GB	352MB	U	
samuelakosaonyekwe/docker_demo:v1	09a965eed2d4	312MB	89.1MB	U	
sleepvsepoint:latest	64f42ce38f89	13MB	3.87MB		

```

● ubuntu@ip-172-31-21-127:~/lab2-task3$ docker run -d --name example sleepvsepoint:latest
e519cce709266fdec3825ada9c5d99488fbcc455cedd4936e4a7085f5cf2ef6e
● ubuntu@ip-172-31-21-127:~/lab2-task3$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	NAMES
e519cce70926	sleepvsepoint:latest	"sleep 10"	About a minute ago	example
Exited (0) About a minute ago				
a65a72404f83	samuelakosaonyekwe/docker_demo:v1	"apache2ctl -DFOREG..."	15 minutes ago	
Up 15 minutes		8080/tcp, 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp		my-container
b9b72820fd18	myapp	"docker-entrypoint.s..."	About an hour ago	
Exited (137) About an hour ago				myapp-containe

```

r
● ubuntu@ip-172-31-21-127:~/lab2-task3$ docker run -d --name example2 sleepvsepoint:latest 30
f698c38c2415cf051fafe596043d7a08d764efb9e5f00b2c72f2edbce3e9648f
● ubuntu@ip-172-31-21-127:~/lab2-task3$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STAT
f698c38c2415	sleepvsepoint:latest	"sleep 30"	24 seconds ago	Up 2
4 seconds				example2
e519cce70926	sleepvsepoint:latest	"sleep 10"	2 minutes ago	Exit
ed (0) 2 minutes ago				example

```
ubuntu@ip-172-31-21-127:~/lab2-task3$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STAT
US	PORTS		NAMES	
f698c38c2415	sleepvsepoint:latest	"sleep 30"	24 seconds ago	Up 2
4 seconds			example2	
e519cce70926	sleepvsepoint:latest	"sleep 10"	2 minutes ago	Exit
ed (0) 2 minutes ago			example	
a65a72404f83	samuelakosaonyejekwe/docker_demo:v1	"apache2ctl -DFOREGR..."	15 minutes ago	Up 1
5 minutes		8080/tcp, 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	my-container	
b9b72820fd18	myapp	"docker-entrypoint.s..."	2 hours ago	Exit
ed (137) About an hour ago			myapp-container	

```
ubuntu@ip-172-31-21-127:~/lab2-task3$
```