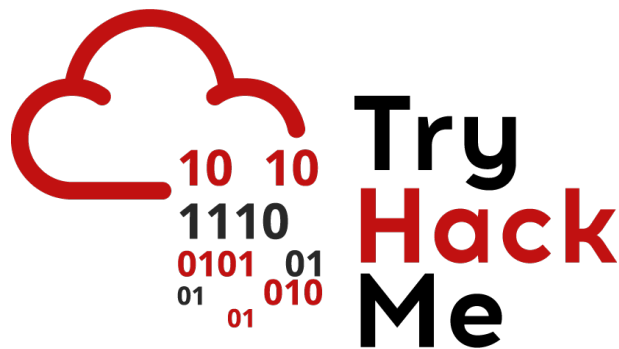


Writeup: Sala *Intro to Docker*

Autor: Ismaeldevs

Plataforma: TryHackMe

4 de julio de 2025



Índice

1. Introducción	2
2. Sala	2
2.1. Tarea 1 – Introducción	2
2.2. Tarea 2 - Sintaxis básica de Docker	2
2.3. Tarea 3 - Poniendo en funcionamiento tu primer contenedor	3
2.4. Tarea 4 - Introducción a Dockerfiles	3
2.5. Tarea 5 - Introducción a Docker Compose	4
2.6. Tarea 6 - Introducción al Docker Socket	5
2.7. Tarea 7 - Práctica	5
3. Conclusión sobre la Sala	8

1. Introducción

Esta sala nos brinda una comprensión inicial pero sólida del funcionamiento y uso de Docker, una herramienta esencial en entornos de desarrollo, despliegue y DevOps. Aprenderemos desde los conceptos más básicos como la sintaxis de comandos y el manejo de contenedores, hasta temas más avanzados como la creación de imágenes con Dockerfiles, servicios con Docker Compose y la interacción a través del Docker socket.

2. Sala

2.1. Tarea 1 – Introducción

En esta primera tarea nos presenta la sala y se explica qué objetivos y tareas se van abordar.

Pregunta: Complete this question before progressing to the next task.

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.2. Tarea 2 - Sintaxis básica de Docker

Aprenderemos los comandos esenciales de Docker para administrar imágenes, cómo descargar imágenes (**docker pull**), listar imágenes (**docker image ls**), y trabajar con etiquetas. También, aprendemos a eliminar imágenes (**docker image rm**) y usos generales del prefijo **docker image**

Una vez aprendemos estos comandos esenciales, procedemos a responder las siguientes preguntas.

Pregunta: If we wanted to pull a docker image, what would our command look like?

Respuesta: **docker pull**

Pregunta: If we wanted to list all images on a device running Docker, what would our command look like?

Respuesta: **docker image ls**

Pregunta: Let's say we wanted to pull the image tryhackme (no quotations); what would our command look like?

Respuesta: **docker pull tryhackme**

Pregunta: Let's say we wanted to pull the image tryhackme with the tag "1337"(no quotations). What would our command look like?

Respuesta: `docker pull tryhackme:1337`

2.3. Tarea 3 - Poniendo en funcionamiento tu primer contenedor

Vamos a aprender a iniciar contenedores desde una imagen. Nos explica los modos interactivo (**-it**), modo desatendido (**-d**), así como asignación de puertos (**-p**). Además, enseña cómo supervisar los contenedores con **docker ps** para activos y **docker ps -a** para todos.

Ahora que entendemos como iniciar contenedores, procedemos a responder las siguientes preguntas.

Pregunta: What would our command look like if we wanted to run a container interactively?

Nota: Assume we are not specifying any image here.

Respuesta: `docker run -it`

Pregunta: What would our command look like if we wanted to run a container in detached mode?

Nota: Assume we are not specifying any image here.

Respuesta: `docker run -d`

Pregunta: Let's say we want to run a container that will run and bind a webserver on port 80. What would our command look like?

Nota: Assume we are not specifying any image here.

Respuesta: `docker run -p 80:80`

Pregunta: How would we list all running containers?

Respuesta: `docker ps`

Pregunta: Now, how would we list all containers (including stopped)?

Respuesta: `docker ps -a`

2.4. Tarea 4 - Introducción a Dockerfiles

Veremos qué es un Dockerfile y cómo estructura una imagen. Aprenderemos instrucciones clave como **FROM**, **RUN**, **WORKDIR**, **EXPOSE**, **CMD**. También, entenderemos

mos cómo construir una imagen usando **docker build -t <nombre>**. Además, buenas prácticas para optimizar archivos y reducir la cantidad de capas.

Después de entender Dockerfiles, vamos a responder las siguientes preguntas.

Pregunta: What instruction would we use to specify what base image the container should be using?

Respuesta: **FROM**

Pregunta: What instruction would we use to tell the container to run a command?

Respuesta: **RUN**

Pregunta: What docker command would we use to build an image using a Dockerfile?

Respuesta: **build**

Pregunta: Let's say we want to name this image; what argument would we use?

Respuesta: **-t**

2.5. Tarea 5 - Introducción a Docker Compose

En esta tarea se nos presenta **Docker Compose** para definir y ejecutar múltiples contenedores mediante un archivo YAML (**docker-compose.yml**). Nos detallan algunos comandos como **docker-compose up** y **docker-compose down** para arrancar y detener servicios.

En base lo aprendido de Docker Compose, responderemos las siguientes preguntas.

Pregunta: I want to use docker-compose to start up a series of containers. What argument allows me to do this?

Respuesta: **up**

Pregunta: I want to use docker-compose to delete the series of containers. What argument allows me to do this?

Respuesta: **down**

Pregunta: What is the name of the .yaml file that docker-compose uses?

Nota: for this question, you will need to include the .yaml file extension in your answer

Respuesta: **docker-compose.yml**

2.6. Tarea 6 - Introducción al Docker Socket

En esta tarea se introduce el socket de Docker (**docker.sock**) y se describe cómo el daemon actúa como una **API IPC**. Además, se define qué es **IPC (Inter-process Communication)** y cómo Docker permite interacción entre procesos mediante ese mecanismo.

Una vez entendido el socket de Docker, procederemos a responder las siguientes preguntas.

Pregunta: What does the term IPC stand for?

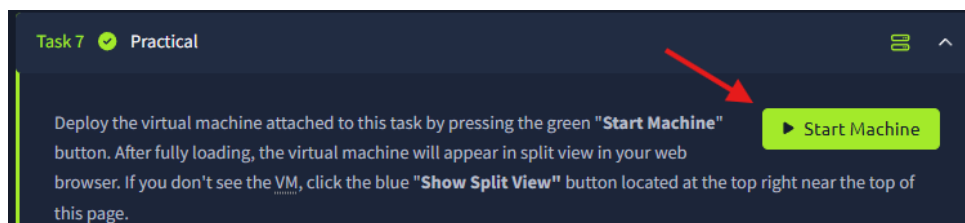
Respuesta: **Interprocess Communication**

Pregunta: What technology can the Docker Server be equalled to?

Respuesta: **API**

2.7. Tarea 7 - Práctica

Ahora utilizaremos los conocimientos anteriores para completar esta tarea. Antes de comenzar, vamos a inicializar la máquina virtual haciendo clic en **Start Machine** en el lado superior.



```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1014-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu Jul  3 02:49:17 UTC 2025

System load:  1.62               Processes:           129
Usage of /:   14.5% of 21.26GB   Users logged in:    0
Memory usage: 17%               IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                IPv4 address for ens5:  10.10.190.106

156 updates can be applied immediately.
91 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

cmnatic@thm-intro-to-docker:~$
```

Una vez que iniciamos la máquina y nos encontramos en ella, procederemos a responder las siguientes preguntas.

Pregunta: Connect to the machine. What is the name of the container that is currently running?

Debemos identificar el contenedor activo en la máquina, para ello, utilizaremos el comando:

■ docker ps

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1014-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu Jul  3 02:52:05 UTC 2025

System load:  1.26               Processes:           125
Usage of /:   14.4% of 21.26GB   Users logged in:    0
Memory usage: 11%               IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                IPv4 address for ens5:  10.10.51.92

156 updates can be applied immediately.
91 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

cmnatic@thm-intro-to-docker:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
0f24bed8b5ef   cloudisland "tail -f /dev/null"     2 years ago Up 43 seconds         CloudIsland
cmnatic@thm-intro-to-docker:~$
```

Respuesta: CloudIsland

Pregunta: Use Docker to start a web server with the webserver image (no quotations). You will need to run the container with port 80.

After starting the container, try to connect to https://LAB_WEB_URL.p.thmlabs.com/ in your browser. What is the flag?

Ahora tenemos que desplegar un contenedor web usando la imagen **webserver**, exponiendo el puerto **80**. Para ello, utilizaremos el siguiente comando

■ docker run -d -p 80:80 webserver

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1014-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Thu Jul  3 02:55:34 UTC 2025

System load:  1.18           Processes:           122
Usage of /:   14.4% of 21.26GB Users logged in:     0
Memory usage: 10%           IPv4 address for ens5: 10.10.165.125
Swap usage:   0%

156 updates can be applied immediately.
91 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

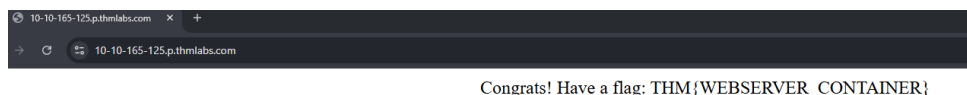
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

cmnatic@thm-intro-to-docker:~$ docker run -d -p 80:80 webserver
73e519dcf13292ee4c4d8919801c0aa400f41ce4aa9cf3de817894c826b44047
cmnatic@thm-intro-to-docker:~$
```

Una vez desplegado el contenedor web, vamos a intentar conectarnos al sitio y lograr obtener la flag.

**Respuesta: THM{WEBSERVER_CONTAINER}**

3. Conclusión sobre la Sala

Al finalizar la sala, habremos logrado comprender los fundamentos de Docker, desde cómo obtener y ejecutar contenedores hasta cómo construir imágenes propias y gestionar entornos multi-contenedor. Además, cómo Docker se comunica internamente mediante el socket, lo cual es clave para entender su arquitectura.