Writeup: Sala *Intro to Containerisation*

Autor: Ismaeldevs

Plataforma: TryHackMe

4 de julio de 2025

Índice

1. Introducción

En la sala aprenderemos qué es la containerización, cómo funciona y por qué se ha convertido en una herramienta esencial en el desarrollo y despliegue de aplicaciones. Además, nos presenta a Docker, la plataforma más popular para trabajar con contenedores, abordando su historia, sus características clave y cómo utilizarla para construir y ejecutar contenedores.

2. Sala

2.1. Tarea 1 – Introducción

En esta primera tarea nos presenta los resultados del aprendizaje al completar la sala como: entender qué es la containerización, por qué se utiliza, comprender cómo funciona esta tecnología.

Pregunta: Complete this question and progress on to the next task.
Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.2. Tarea 2 - ¿Qué es la contenerización?

Aprenderemos que la containerización consiste en empacar una aplicación junto con sus dependencias (librerías, frameworks) en un contenedor portátil. Además, destaca cómo este enfoque resuelve problemas de compatibilidad y facilita entornos consistentes. Por último nos menciona la funcionalidad del kernel namespace para aislar procesos, evitando que interactúen entre sí.

Ahora que comprendemos qué es la contenerización, procedemos a responder las siguientes preguntas.

Pregunta: What is the name of the kernel feature that allows for processes to use resources of the Operating System without being able to interact with other processes?
Respuesta: **namespace**

Pregunta: In a normal configuration, can other containers interact with each other? (yay/nay)
Respuesta: **nay**

2.3. Tarea 3 - Presentando Docker

Se nos presenta **Docker** como plataforma para crear, distribuir y ejecutar contenedores. Esta define que al publicarse una aplicación pasa a ser una imagen y se utiliza YAML para que los desarrolladores puedan indicar cómo se debe construir un contenedor y qué se debe ejecutar. También, nos comenta la función del **Docker Engine**, que permite gestionar imágenes, interconectar contenedores y facilitar el intercambio de artefactos.

Ahora que conocemos Docker, podemos responder las siguientes preguntas.

Pregunta: What does an application become when it is published using Docker? Format: An xxxxx (fill in the x's)

Respuesta: An Image

Pregunta: What is the abbreviation of the programming syntax language that Docker uses?

Respuesta: YAML

2.4. Tarea 4 - La historia de Docker

Conoceremos el origen de Docker, creado por Solomon Hykes dentro de dotCloud en 2013 y presentado en PyCon ese mismo año. Los rudimentos de los contenedores ya existían en UNIX V7 (1979), pero Docker democratizó y modernizó la tecnología.

Después de conocer un poco la historia de Docker, responderemos las siguientes preguntas.

Pregunta: In what year was Docker originally created?
Respuesta: 2013

Pregunta: Where was Docker first showcased?
Respuesta: Pycon

Pregunta: What version of Unix had the first concepts of containerisation?
Respuesta: V7

2.5. Tarea 5 - Los beneficios y características de Docker

En esta tarea aprenderemos las ventajas de Docker, las cuales nos comenta que es gratuito, multiplataforma, eficiente en recursos (usa menos espacio y RAM comparado con las VM), de fácil adopción gracias a su documentación y comunidad, portátil

gracias a las imágenes, seguro por su diseño minimalista y en entornos Cloud más económico.

Pregunta: Read me!
Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.6. Tarea 6 - ¿Cómo funciona la contenerización?

Ahora profundizaremos en el mecanismo técnico, donde cada contenedor corre en su propio namespace, por lo que solo ve los procesos dentro de su ámbito. Nos ilustran cómo el host y los contenedores tienen sus propios PIDs aislados, explicando cómo esto proporciona aislamiento (y los riesgos si no se aísla correctamente).

Después de comprender como funciona la contenerización, procederemos a responder la siguiente pregunta.

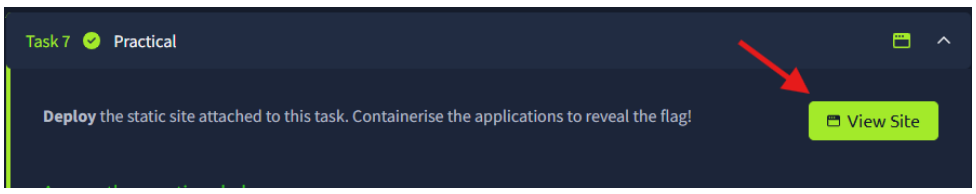
Pregunta: What command can we use to view a list of running processes?
Respuesta: **ps aux**

2.7. Tarea 7 - Práctica

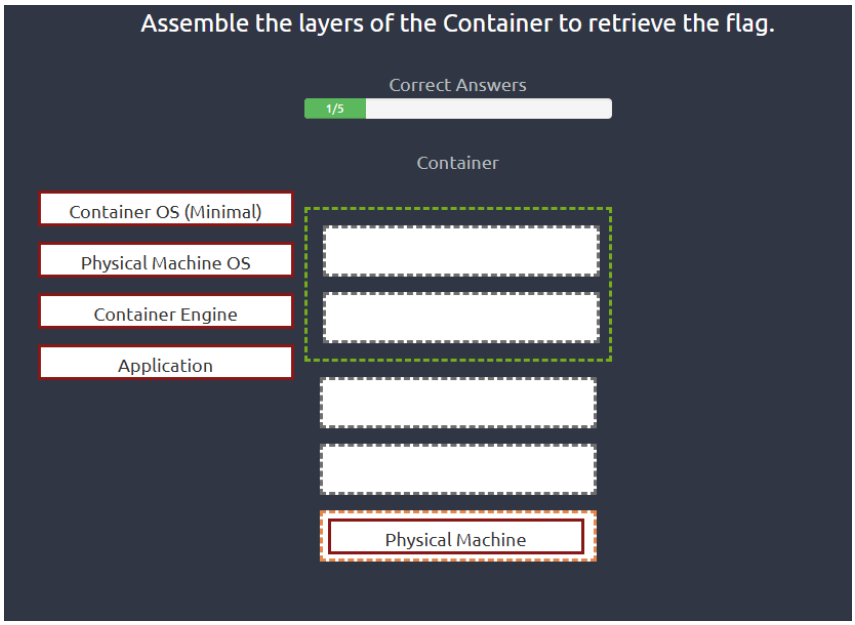
Ahora debemos containerizar una aplicación estática utilizando Docker para revelar la flag. Es decir, construir y ejecutar un contenedor que despliegue el sitio.

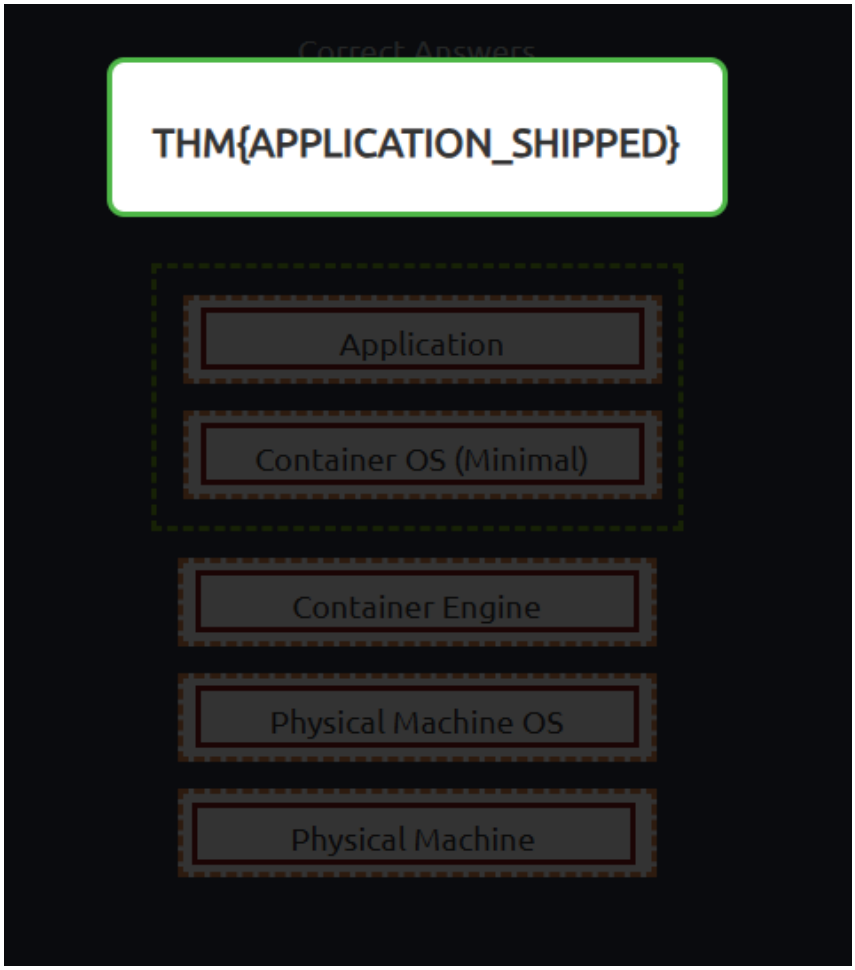
Pregunta: Containerise the applications in the static site. What is the flag?

Para comenzar, vamos a desplegar el sitio de práctica haciendo clic en **View Site** en el lado superior de la tarea.



Una vez desplegado el sitio, procederemos a ensambla las capas del contenedor para obtener la flag.





Respuesta: **THM{APPLICATION_SHIPPED}**

3. Conclusión sobre la Sala

Al finalizar la sala, logramos entender los principios básicos de la containerización y su aplicación en entornos reales utilizando Docker. También, Aprendimos tanto los aspectos teóricos como técnicos que permiten a los contenedores funcionar de forma aislada, segura y eficiente.