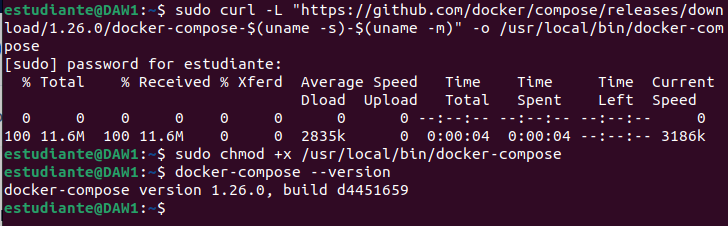
Actividad 11-10-2022

**1. ¿Que es Docker Compose? ¿Qué utilidades tiene? Instálalo y comprueba que está instalado**

Docker Compose es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores, usa un archivo YAML para configurar los servicios de la aplicación. Después, con un solo comando, se crean y se inician todos los servicios de la configuración



Instalación, aplicación de privilegios de ejecución y verificación de su instalación.

2. Haz un glosario de comandos de Docker Compose

* **docker-compose build:** Se usa para generar la imagen, basada en las especificaciones del servicio en el docker-compose
* **docker-compose up:** En caso de no existir la imagen (build previo), se arma la misma, y siempre se crea el container y luego se inicia.
* **docker-compose start:** Se inicia un container existente
* **docker-compose stop:** Frena un container, sin eliminarlo, para luego poder iniciarse nuevamente con start.
* **docker-compose down:** Se utiliza para frenar el/los container/s, eliminarlos.
* **docker-compose ps:** Lista los containers corriendo en el contexto de este docker-compose.
* **docker-compose.yml:** es el archivo donde residen las instrucciones y configuraciones de él/los servicios.

**3. ¿Qué es un fichero YAML? ¿Qué estructura tiene? Investiga y detállalo. Haz también un glosario de YAML.**

Este formato de serialización de datos se encarga de almacenar archivos de configuración y se puede usar junto con todos los lenguajes de programación. YAML no propone etiquetas solo formato e identificadores mínimos, centrándose realmente en los datos.

Cada YAML comienza con --- que denota el inicio de un archivo YAML.

Los strings no deben estar entre comillas, aunque también es válido.

Los strings de líneas múltiples pueden ser escritos como un 'bloque literal' (usando pipes |) o como un 'bloque doblado' (usando >)

La indentación se usa para anidar elementos:

**un\_mapa\_indentado:**

**llave: valor**

**otra\_llave: otro valor**

**otro\_mapa\_indentado:**

**llave\_interna: valor\_interno**

Las llaves de los mapas no requieren ser strings necesariamente.

0.25: una llave numérica

Las colecciones en YAML usan la indentación para delimitar el alcance y cada elemento de la colección inicia en su propia línea.

Las secuencias (equivalentes a listas o arreglos) se ven así:

**- Amarillo**

**- Verde**

**- Azul**

Las secuencias pueden tener distintos tipos en su contenido.

**secuencia\_combinada:**

**- texto**

**- 5**

**- 0.6**

**- llave: valor**

**-**

**- Esta es una secuencia**

**- ...dentro de otra secuencia**

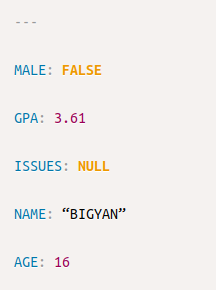
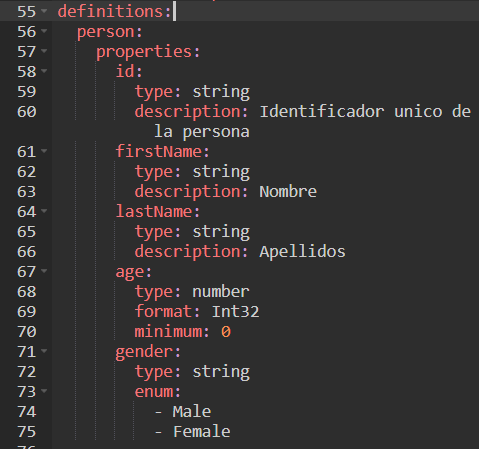
Dado que todo JSON está incluído dentro de YAML, también puedes escribir mapas con la sintaxis de JSON y secuencias:

mapa\_de\_json\_1: {"llave": "valor"}

mapa\_de\_json\_2:

llave: valor

Ejemplos:

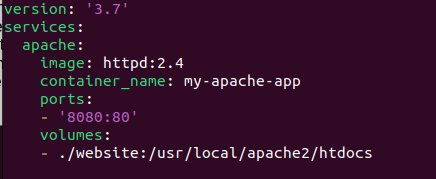


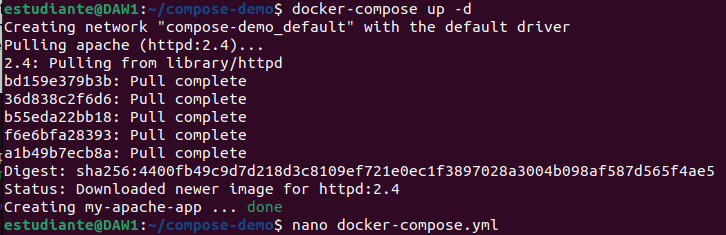
**4. Crea un archivo de configuración para una aplicación que contiene un único servicio:**

**- Imagen: httpd:2.4 (del servidor web Apache)**

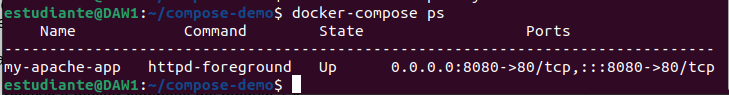
**- Puerto: el host de Docker publicará el puerto 80 y hará una redirección con el puerto 80 del contenedor.**

**- Bind mount: lo creamos entre el directorio actual del host de Docker y el directorio *usr*/local/apache2/htdocs/ del contenedor (que es el directorio que utiliza el servidor web para servir el contenido que encuentre en su interior.**





**5. Consulta la lista de contenedores que están en ejecución y explica la salida. ¿Qué diferencia hay entre “docker ps y docker compose ps”?**



**Name:** Indica el nombre de nuestra app.

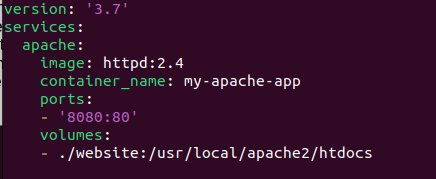
**Command:** Indica el tipo de servidor web que estamos utilizando en este caso utilizamos httpd de apache.

**State:** Indica el estado de nuestra app.

**Ports:** Nos aparece los puertos utilizados y los redireccionamientos del mismo.

**docker ps:** Muestra todos los contenedores en ejecución en el motor docker. **docker-compose ps:** Muestra los contenedores relacionados con las imágenes declaradas en docker-compose file.

**6. Muestra la salida de docker-compose.yml**



**7. Visiona el video relativo a docker compose del módulo 7.4.**

HECHO