

Informe Técnico: Semana 12

Carolina Tobaría¹, Andrés Bravo², Jeremías Pabón³

¹⁻²Escuela de Ingeniería, ³Dpto. de Comunicación,

Universidad Central

Maestría de Analítica de Datos

Curso Automatización e Integración de datos para la inteligencia artificial

Bogotá, Colombia

¹ctobariac@ucentral.edu.co, ²abravop@ucentral.edu.co, ³jpabonl@ucentral.edu.co

May 23, 2024

Contents

1	Resumen o resumen ejecutivo	2
2	Introducción	2
3	Contexto y antecedentes	3
4	Metodología	4
4.1	Objetivos del Taller	4
4.2	Requisitos	4
4.3	Materiales	4
4.4	Introducción a Docker	4
4.4.1	Imágenes y Contenedores	4
4.4.2	Línea de Comandos de Docker	4
4.5	Ejecutando una Imagen de Docker Preexistente	5
4.6	Creación de una Imagen de Docker para una Aplicación Python	5
5	Conclusión	6
6	Observaciones	6

1 Resumen o resumen ejecutivo

A continuación presentamos el resumen ejecutivo del informe sobre la clase de Inteligencia Artificial del 11/05/2024:

Se da inicio a la clase donde se socializa con algunos compañeros las noticias relevantes de la semana sobre tecnología.

Los talleres de Docker en línea se centran en brindar una introducción práctica a los conceptos fundamentales de Docker, tales como imágenes, contenedores y la línea de comandos de Docker. Durante las sesiones, los participantes aprenden a ejecutar una imagen de Docker preexistente y a crear su propia imagen Docker para una aplicación Python sencilla. Estos talleres, realizados en una plataforma interactiva como Play with Docker, ofrecen una experiencia práctica y guiada para familiarizarse con Docker y su funcionamiento.

En el primer taller, los participantes aprenden a ejecutar una imagen de Docker preexistente utilizando el comando `docker run`. Se les guía a través de los pasos para acceder a Play with Docker, crear una instancia de Docker, abrir una terminal y ejecutar comandos para lanzar un contenedor a partir de una imagen de Ubuntu preexistente. Este ejercicio proporciona una introducción práctica a los conceptos básicos de Docker y su funcionamiento.

En el segundo taller, los participantes crean su propia imagen Docker para una aplicación Python sencilla. Se les enseña cómo escribir un archivo `Dockerfile` con las instrucciones necesarias para construir la imagen, incluyendo la especificación de la imagen base, la instalación de dependencias y la configuración del comando de inicio. Luego, utilizan el comando `docker build` para construir la imagen y el comando `docker run` para ejecutar un contenedor a partir de ella. Este taller ofrece una visión más profunda de cómo crear y gestionar imágenes personalizadas en Docker.

En resumen, los talleres de Docker en línea proporcionan una introducción práctica y paso a paso a los conceptos básicos de Docker, permitiendo a los participantes adquirir habilidades fundamentales para trabajar con esta tecnología en entornos de desarrollo y despliegue de aplicaciones.

2 Introducción

Docker ha emergido como una herramienta fundamental en el mundo del desarrollo de software, permitiendo a los equipos construir, distribuir y ejecutar aplicaciones de manera consistente y eficiente en entornos virtualizados. Su adopción ha crecido rápidamente debido a su capacidad para simplificar el proceso de desarrollo y despliegue de aplicaciones, así como para facilitar la administración de infraestructuras de TI.

En este contexto, este informe se enfoca en presentar una visión integral de los talleres de Docker en línea, los cuales ofrecen una introducción práctica a los conceptos fundamentales de Docker. Estos talleres no solo abordan los aspectos técnicos de Docker, como la creación y ejecución de contenedores, sino que también exploran su relevancia filosófica y práctica en el mundo actual de

la tecnología.

El objetivo de este informe es proporcionar una visión completa de los talleres de Docker en línea, destacando su importancia en el panorama actual de desarrollo de software. Se abordarán los conceptos básicos de Docker, los principios filosóficos que subyacen a su tecnología, así como las implicaciones prácticas de su uso en entornos de desarrollo y despliegue de aplicaciones. Este informe servirá como una guía introductoria para aquellos que deseen familiarizarse con Docker y su papel en el desarrollo de software moderno.

3 Contexto y antecedentes

Qué es un contenedor en computación y cómo funciona

Los contenedores en computación son entornos de ejecución estandarizados y livianos que encapsulan una aplicación y todas sus dependencias, incluidas bibliotecas y archivos de configuración, en una unidad única y portátil. Funcionan mediante la virtualización a nivel de sistema operativo, lo que significa que comparten el núcleo del sistema operativo del host, pero se ejecutan de manera aislada y sin interferencias entre sí. Esto permite que los contenedores sean altamente portables y eficientes, ya que pueden ejecutarse en cualquier entorno que admita contenedores, independientemente de la infraestructura subyacente.

Qué es Docker y cómo funciona:

Docker es una plataforma de código abierto que facilita la creación, distribución y ejecución de contenedores de software. Funciona mediante el uso de una serie de herramientas y tecnologías, incluido el motor de contenedores Docker, que administra la creación y ejecución de contenedores; Docker Compose, que permite definir y ejecutar aplicaciones multi-contenedor; y Docker Swarm, que facilita el escalado y la administración de aplicaciones en un clúster de contenedores. Docker utiliza un enfoque basado en imágenes para definir el entorno de ejecución de una aplicación, lo que permite a los desarrolladores crear y compartir imágenes de contenedores fácilmente a través del Docker Hub o repositorios privados.

Tipos de máquinas

Finalmente, se establece una categorización de máquinas físicas, biológicas y lógicas; resaltando las capacidades complementarias entre humanos y sistemas artificiales, así como la importancia de un desarrollo mancomunado de la inteligencia biológica y la computacional.

Espero que esta ampliación de la sección de Contexto y Antecedentes sirva para enriquecer las bases conceptuales presentadas en la clase en torno a la inteligencia artificial y su interrelación con la condición humana desde una perspectiva multilayered.

4 Metodología

Dado que los estudiantes tienen distintos niveles de conocimiento sobre Docker y programación en general, el profesor nos enseñará a utilizar herramientas prácticas y accesibles como "Play with Docker" para aprender y experimentar con los conceptos. Este enfoque permitirá a los estudiantes materializar rápidamente los conocimientos adquiridos en un entorno controlado y sin necesidad de configuraciones complejas en sus propios equipos.

4.1 Objetivos del Taller

El taller tuvo los siguientes objetivos:

1. Aprender los conceptos básicos de Docker, incluyendo imágenes, contenedores y la línea de comandos de Docker.
2. Ejecutar una imagen de Docker preexistente.
3. Crear y ejecutar una imagen de Docker para una aplicación Python sencilla.

4.2 Requisitos

Para participar en el taller, los estudiantes necesitaron:

- Un navegador web.

4.3 Materiales

El taller utilizó la plataforma "Play with Docker", accesible en url : <https://labs.play-with-docker.com/>.

4.4 Introducción a Docker

Docker es una plataforma de contenedores que permite empaquetar aplicaciones y sus dependencias en unidades de software estandarizadas y ligeras llamadas contenedores. Los contenedores se ejecutan en un entorno aislado, lo que los hace muy portables y eficientes.

4.4.1 Imágenes y Contenedores

- ****Imagen de Docker****: Es una plantilla que contiene todo lo necesario para ejecutar una aplicación, incluyendo el sistema operativo, archivos de aplicación y dependencias. - ****Contenedor****: Es una instancia de una imagen de Docker que se ejecuta en un entorno aislado.

4.4.2 Línea de Comandos de Docker

La línea de comandos de Docker permite interactuar con Docker para crear, ejecutar, administrar y eliminar imágenes y contenedores.

4.5 Ejecutando una Imagen de Docker Preexistente

1. Accede a [urlhttps://labs.play-with-docker.com/](https://labs.play-with-docker.com/).
2. Haz clic en el botón "Launch" para crear una nueva instancia de Docker.
3. Una vez creada la instancia, haz clic en el botón "Terminal".
4. En la terminal, ejecuta el siguiente comando para ejecutar la imagen de Ubuntu:

```
docker run ubuntu
```

5. Para ver el estado del contenedor, ejecuta:

```
docker ps
```

4.6 Creación de una Imagen de Docker para una Aplicación Python

1. Accede a [urlhttps://labs.play-with-docker.com/](https://labs.play-with-docker.com/).
2. Haz clic en el botón "Launch" para crear una nueva instancia de Docker.
3. Una vez creada la instancia, haz clic en el botón "Terminal".
4. En la terminal, crea una carpeta para la aplicación Python:

```
mkdir python-app
```

5. En la carpeta `python-app`, crea un archivo `main.py` con el siguiente contenido:

```
print("Hello, world!")
```

6. Crea un archivo `Dockerfile` en la carpeta `python-app` con el siguiente contenido:

```
FROM python:3.10

CMD ["python", "main.py"]
```

7. Para crear la imagen de Docker, ejecuta el siguiente comando:

```
docker build -t python-app .
```

8. Para ejecutar la imagen de Docker, ejecuta:

```
docker run python-app
```

9. Para ver el estado del contenedor, ejecuta:

```
docker ps
```

10. Para detener el contenedor, ejecuta:

```
docker stop <ID_CONTENEDOR>
```

Reemplaza IDCONTENEDOR con el ID del contenedor que deseas detener.

5 Conclusión

En este taller, los estudiantes aprendieron los conceptos básicos de Docker, ejecutando una imagen de Docker preexistente y creando una imagen de Docker para una aplicación Python sencilla. Además, experiencia práctica utilizando "Play with Docker" para manejar contenedores en un entorno web.

6 Observaciones

- "Play with Docker" proporciona una instancia de Docker preconfigurada para probar Docker sin necesidad de instalarlo localmente. - El comando `docker build` tiene muchas opciones que permiten personalizar el comportamiento de la imagen de Docker. Para más información, consulte la documentación oficial de Docker en <https://docs.docker.com/>. - El comando `docker run` también tiene muchas opciones para personalizar el comportamiento del contenedor. Para más detalles, refiérase a la documentación de Docker.