Carrera: Ing. en desarrollo de software.

Materia: Cómputo en la nube.

Trabajo: Docker & Entornos Cloud

Docente: Ing. Contreras Cardona Miguel Angel

Alumnos: Salas Santiz Esman Caciano

Grupo: 03DESVA

Docker & Entornos Cloud

1. Introducción

En la actualidad, la agilidad en el despliegue de software es vital para la competitividad tecnológica. Tecnologías como Docker han transformado la industria al permitir que las aplicaciones se ejecuten en entornos aislados llamados contenedores, eliminando los conflictos de dependencias y optimizando el uso de recursos del sistema en comparación con las máquinas virtuales tradicionales.

En el reporte se documenta el proceso técnico para configurar un nodo interactivo basado en la imagen oficial de Ubuntu. El propósito fundamental es preparar una infraestructura virtualizada y ligera que permita el despliegue del proyecto "Infoga", integrando el uso de GitHub como sistema de control de versiones dentro de un entorno de ejecución controlado y portable.

2. Objetivos

Objetivo General

Configurar un entorno de desarrollo basado en contenedores para el despliegue de herramientas de código abierto, asegurando la portabilidad y el aislamiento de procesos.

Objetivos Específicos

1. **Verificar l**a instalación y operatividad del motor de Docker en el sistema anfitrión para asegurar una base sólida de infraestructura.
2. **Desplegar** un contenedor interactivo utilizando la imagen oficial de Ubuntu para establecer un nodo de trabajo aislado.
3. **Aprovisionar** el entorno interno del contenedor mediante la instalación de dependencias críticas como Git y Python 3.
4. **Integrar** el código fuente del proyecto "Infoga" mediante el uso de herramientas de control de versiones.
5. **Validar** la persistencia de datos y el ciclo de vida del contenedor (start/stop) para garantizar la disponibilidad continua del entorno de trabajo.

3. Desarrollo Técnico

En esta sección se detalla la ejecución cronológica de las actividades, vinculando cada procedimiento con los objetivos específicos del proyecto.

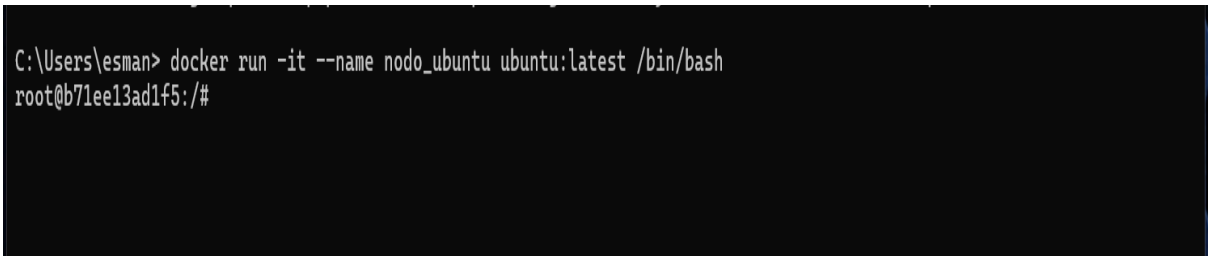
3.1. Preparación y Verificación del Motor de Docker

- **Acción:** Se inició Docker Desktop en el sistema y se procedió a verificar la integridad de la instalación mediante la interfaz de línea de comandos.

- **Comando ejecutado:** docker --version
- **Resultado:** Se confirmó la versión activa de Docker, cumpliendo así con el primer objetivo específico de asegurar la disponibilidad de la plataforma.

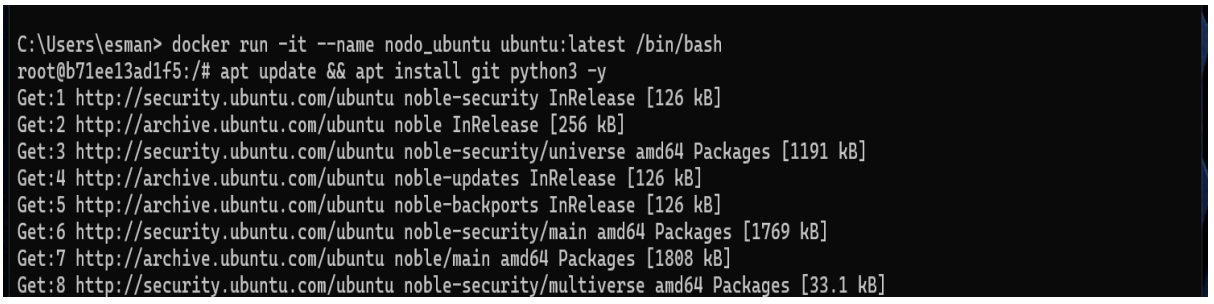
3.2. Aislamiento y Despliegue del Nodo Ubuntu

- **Acción:** Se realizó el despliegue de un contenedor interactivo basado en la imagen oficial más reciente de Ubuntu. Se asignó el nombre nodo_ubuntu para facilitar su gestión posterior.
- **Comando ejecutado:** docker run -it --name nodo_ubuntu ubuntu:latest /bin/bash
- **Resultado:** El sistema creó un entorno aislado y proporcionó acceso inmediato a la terminal del contenedor, cumpliendo con el segundo objetivo específico.



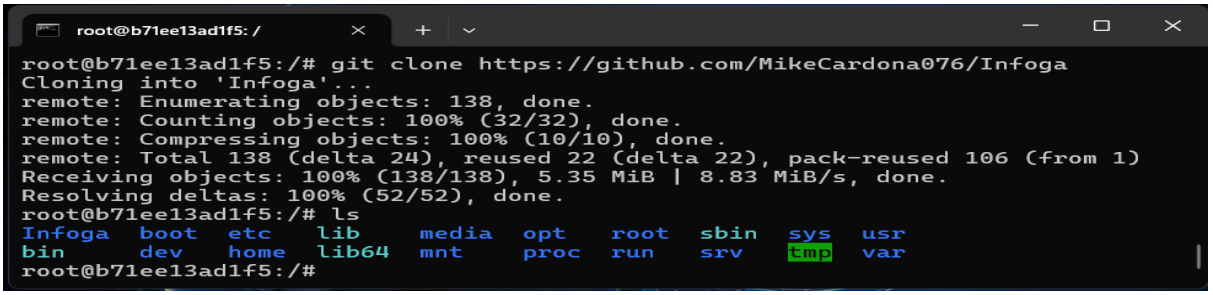
3.3. Aprovisionamiento del Entorno Interno

- **Acción:** Una vez dentro del nodo, se procedió a actualizar los índices de los paquetes e instalar las herramientas necesarias para el desarrollo y ejecución del proyecto.
- **Comando ejecutado:** apt update && apt install git python3 -y
- **Resultado:** Se instalaron con éxito las dependencias de Git y Python 3, cumpliendo con el tercer objetivo específico de configurar el entorno de trabajo.



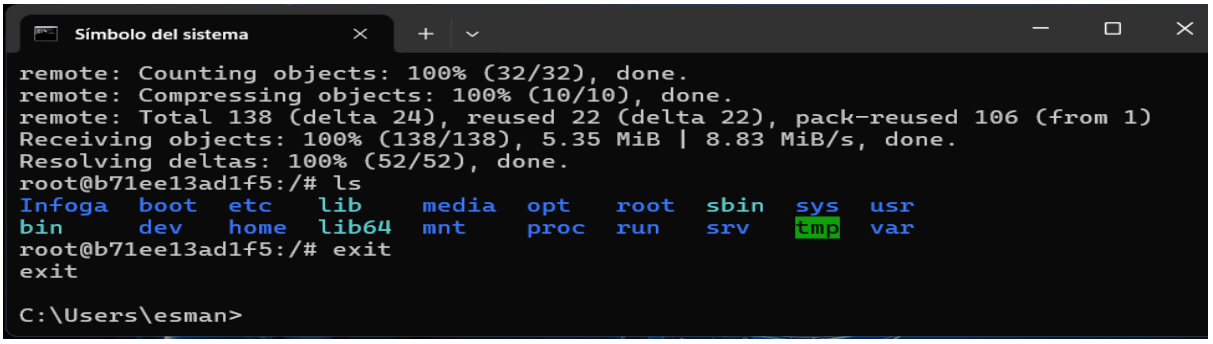
3.4. Integración de Código y Control de Versiones

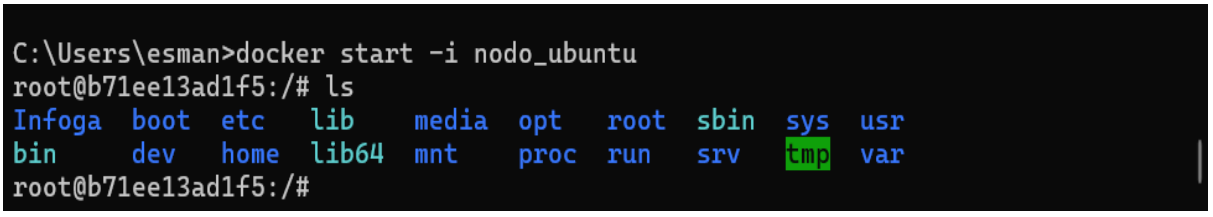
- **Acción:** Utilizando la herramienta Git, se clonó el repositorio remoto "Infoga" para traer el código fuente al entorno local del contenedor.
- **Comando ejecutado:** git clone https://github.com/MikeCardona076/Infoga seguido de ls para verificación.
- **Resultado:** Se validó la presencia de la carpeta del proyecto en el directorio raíz del contenedor, cumpliendo con el cuarto objetivo específico de integración de código fuente.



3.5. Gestión del Ciclo de Vida y Persistencia de Datos

- **Acción:** Se realizó una prueba de detención y reinicio del contenedor para validar que el entorno es persistente y que los cambios realizados como la instalación de software y la clonación de archivos no se pierden al cerrar la sesión.
- **Comandos ejecutados:** exit para salir, seguido de docker start -i nodo_ubuntu para reingresar.
- **Resultado:** Se demostró la capacidad de Docker para mantener el estado del sistema. Al volver a entrar al nodo y ejecutar el comando ls, se confirmó que la carpeta Infoga y las configuraciones de Git/Python permanecieron intactas, cumpliendo con el requisito de evidencia de "Entrar y salir del contenedor (start/stop)".





4. Conclusión Individual

Realizar esta práctica me ayudó a entender de forma real cómo funciona la nube y por qué es tan importante para el desarrollo de software hoy en día. Estas son mis conclusiones sobre el ejercicio:

¿Qué ventajas encontraste al usar contenedores sobre VMs?
Lo que más noté fue la velocidad y la ligereza. Antes pensaba que para tener otro sistema operativo necesitaba una Máquina Virtual (VM) que tardara mucho en prender y que pusiera lenta mi computadora. Con Docker, me sorprendió que el nodo de Ubuntu encendió en segundos. Siento que es mucho más eficiente porque no pesa tanto y no consume toda la memoria de mi PC, lo que me permite trabajar con otras herramientas al mismo tiempo sin problemas.

¿Qué dificultades tuviste al configurar el entorno de Ubuntu?
Me encontré con un par de retos que me enseñaron mucho:

1. **El inicio de Docker:** Al principio, me aparecía un error al querer ver la versión de Docker. Pero luego me di cuenta de que era porque no había abierto la aplicación Docker Desktop. Aprendí que antes de usar la consola, debo asegurarme de que el servicio esté corriendo en mi equipo.
2. **Errores de escritura:** Decidí escribir todos los comandos manualmente en lugar de copiar y pegar para ir memorizándolos. Esto me causó algunos errores de dedo. Aunque fue frustrante en el momento, me sirvió para poner más atención a los detalles y entender qué hace cada parte del comando que estoy enviando.

¿Cómo facilita Docker el despliegue de herramientas como Infoga?
Docker hace que todo sea mucho más sencillo porque evita que te falten archivos o programas. Con Docker, sé que el código está guardado en un contenedor y tiene todo lo que necesita. Si funciona en mi contenedor, sé que funcionará en cualquier otra computadora sin tener que instalar nada más por fuera.