

Tarea Entregable 1:

Ejercicios DOCKERFILE

Realizado por:

GEORGI VANKOV

Ciclo de Desarrollo de Aplicaciones Web

Modalidad Semipresencial

Despliegue de   
Aplicaciones Web

ÍNDICE

1. **Crea un dockerfile basado en ubuntu 20.04 que instale git.**

**2. Crea un dockerfile basado en la última versión de Ubuntu que instalelos paquetes de vim y htop.**

**3. Crea un dockerfile basado en ubuntu 20.04 que instale node.**

**4. Crea un dockerfile que tenga node sin necesidad de instalar Ubuntu.**

**5. Crea un dockerfile basado en alpine en el cual instalemos el servidor**

**web nginx. Tras esto, debe exponerse el puerto 80 y ejecutarse**

**automáticamente.**

**6. Crea un dockerfile basado en alpine en el cual instalemos el servidor web nginx. Debes copiar un fichero index.html el cual tenga el texto “Hola Nginx en Docker”. Tras esto, debe unir los puertos 8081 local al 80 del contenedor.**

1-----------------------------------------------------------------------------------------------

**Captura de pantalla de computadora

Descripción generada automáticamente**

**Podemos ver como estamos ejecutando la consola de Ubuntu y hemos creado un nuevo directorio.**

**También hemos iniciado el contenedor de forma interactiva con el comando -it**

**Texto

Descripción generada automáticamente**

# Utiliza la imagen base de Ubuntu 20.04

FROM ubuntu:20.04

# Actualiza los repositorios e instala Git

RUN apt-get update && apt-get install -y git

# Comando por defecto al iniciar el contenedor

CMD ["bash"]

2------------------------------------------------------------------------------------------------------------------------------------

**Texto

Descripción generada automáticamente**

**Hemos ejecutado el segundo contenedor.**

**Hemos abierto para probar que esta instalado el editor de texto vim**

**Texto

Descripción generada automáticamente**

# Utilizamos imagen base última versión de Ubuntu

FROM ubuntu:latest

# apt-get -y update Actualizamos lista de paquetes

RUN apt-get -y update

# apt-get -y upgrade Actualizamos los paquetes del SO a ultimas versiones

RUN apt-get -y upgrade

# vim instalamos editor de texto , htop instalamos la herramienta

RUN apt-get -y install apt-utils vim htop

# Comando por defecto al iniciar el contenedor

CMD ["bash"]

3----------------------------------------------------------------------------------------------------------------------

Este se me quedaba pillado en el último punto.

# Utiliza la imagen base de Ubuntu 20.04

FROM ubuntu:20.04

# Actualiza lista de paquetes y actualiza paquetes del SO

RUN apt-get update

RUN apt-get upgrade -y

RUN apt-get install -y apt-utils nodejs npm

RUN apt-get clean

# Comando por defecto al iniciar el contenedor

CMD ["bash"]

4---------------------------------------------------------------------------------------------------------------------------------

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Lo he tenido que hacer y rehacer, a la hora de instalar las dependencias solo con /app no me funcionaba solamente creaba un archivo y no lo copiaba dentro.**

**Pero ejecutamos y probamos la versión de Node.js en este caso la 14.21.3**

#Utilizamos la imagen base de Node.js 14 slim

FROM node:14-slim

#Copiamos las dependencias de nuestor json al directorio de Docker

COPY package.json /app/package.json

# Instala las dependencias de la aplicación (si es necesario)

RUN npm install

# Comando por defecto al iniciar el contenedor

CMD ["npm", "start"]

5-------------------------------------------------------------------------------------------------------------------------------------

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Entramos a localhost80 y el que hemos expuesto en Dockerfile es el 80**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

# Utiliza la imagen base de Alpine Linux

FROM alpine:latest

# Actualiza el sistema y luego instala Nginx

RUN apk update && apk upgrade && apk add nginx

# Exponemos el puerto 80 para que Nginx pueda ser accesible

EXPOSE 80

# Comando para iniciar Nginx automáticamente cuando se ejecute el contenedor

CMD ["nginx", "-g", "daemon off;"]

6----------------------------------------------------------------------------------------------------------------------------------------------------------------

Este entraba el nginx pero por algún tema de las rutas o algo no encontraba el archivo html

# Utiliza la imagen base de Alpine Linux

FROM alpine:latest

# Copiamos el archivo html

COPY index.html /usr/share/nginx/html/index.html

# Actualiza el sistema y luego instala Nginx

RUN apk update && apk upgrade && apk add nginx

# Exponer el puerto 80 para que Nginx sea accesible

EXPOSE 80

# Comando para iniciar Nginx automáticamente cuando se ejecute el contenedor

CMD ["nginx", "-g", "daemon off;"]

1.- Ejercicio 1

2.- Ejercicio 2